

Sensor-based Detection and Classification of Soccer Goalkeeper Training Exercises

JUAN HALADJIAN, Technical University Munich

DANIEL SCHLABBERS, Technical University Munich

SAJJAD TAHERI, Technical University Munich

MAX THARR, Technical University Munich

BERND BRUEGGE, Technical University Munich

Many goalkeeper trainees cannot afford a personal human coach. Hence, they could benefit from a virtual coach that provides personalized feedback about the execution of their training exercises. As a first step towards this goal, we developed an algorithm to detect and classify goalkeeper training exercises using a wearable inertial sensor attached to a goalkeeper glove. We collected data from 14 goalkeeper trainees while performing a series of training exercises (e.g., dives, catches, throws). Our approach first detects the exercises using an event detection algorithm based on a high-pass filter, a peak detector, and Dynamic Time Warping to detect and eliminate irrelevant motion instances. Then, it extracts a set of statistical and heuristic features to describe the different exercises and train a machine learning classifier. Our exercise detection approach retrieves 93.8% of the relevant exercises with 90.6% precision and classifies the detected exercises with an accuracy of 96.5%. The exercises recognized by our algorithm can be used to compute further qualitative metrics about individual exercise executions to provide goalkeepers with relevant feedback about their training.

CCS Concepts: • **Human-centered computing** → *Personal digital assistants*; • **Hardware** → *Sensor devices and platforms*;

Additional Key Words and Phrases: Soccer, goalkeeping, event detection, wearable sensor, activity recognition, signal processing, machine learning, dynamic time warping

ACM Reference Format:

Juan Haladjian, Daniel Schlabbers, Sajjad Taheri, Max Tharr, and Bernd Bruegge. 2018. Sensor-based Detection and Classification of Soccer Goalkeeper Training Exercises. *ACM Trans. Internet Things* 1, 1, Article 1 (November 2018), 20 pages. <https://doi.org/0000001.0000001>

1 INTRODUCTION

Soccer is one of the most popular sports in the world with numerous professional and an even larger number of non-professional practitioners. A soccer team consists of ten field players and a goalkeeper. Due to their unique role within the team, goalkeepers undergo a different training than the rest of the players in the team. Goalkeepers train a specific set of well-defined motions to consolidate them into muscle memory and lower their reaction time during

Authors' addresses: Juan Haladjian, Technical University Munich, juan.haladjian@tum.de; Daniel Schlabbers, Technical University Munich, daniel.schlabbers@tum.de; Sajjad Taheri, Technical University Munich, sajjad.taheri@tum.de; Max Tharr, Technical University Munich, max.tharr@tum.de; Bernd Bruegge, Technical University Munich, bruegge@tum.de.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

Manuscript submitted to ACM



Fig. 1. Goalkeeper training exercise variations.

a game. The correct execution of the exercises have a steep learning curve and can only be assessed by experienced trainers.

Despite the importance of personalized feedback from a professional trainer, only older, more experienced goalkeepers enrolled in a soccer club have access to a dedicated trainer. In contrast, most young goalkeeper trainees cannot afford a personalized coach. The ultimate goal of our work is to realize a system based on an unobtrusive wearable sensor able to provide personalized and objective feedback to goalkeepers automatically after or during a training session to help them improve their skills.

A goalkeeper training session typically includes exercises to stop incoming balls (e.g., dives, catches, jumps) and others to pass the ball at field players. Goalkeepers execute different variations of these exercises to learn the right

105 movements—mostly involving the legs, torso and arms—to handle each incoming ball. For example, they execute a
106 different motion to catch balls thrown at their chest, belly and legs. They catch balls thrown at the chest with their
107 hands, wrap balls thrown at their belly between the hands and their chest and control lower balls on the ground using
108 their entire body. Figure 1 shows different variations of goalkeeper training exercises.
109

110 In this article, we present a smart glove and algorithm to automatically detect and keep track of goalkeeper training
111 exercises using a single inertial sensor. To the best of our knowledge, this application has not been studied previously.
112 The work we present enables goalkeepers to keep track of the training exercises they perform. Goalkeepers could get
113 an overview of the exercises performed by goalkeepers during the last days, weeks or months, which they could use to
114 plan future training sessions accordingly. More importantly, the ability to detect and classify training exercises is a first
115 step towards a virtual coach that extracts performance metrics from the recognized exercises and gives personalized
116 feedback to goalkeepers.
117

118 The high variability in the training exercises and individual execution of each exercise make this application extremely
119 challenging. A particular challenge we address is on filtering out irrelevant motions that originate from the high degree
120 of freedom in the movement of a goalkeeper’s hands. The development process and methods we describe in this article
121 can be reused in similar activity recognition applications.
122

123 This paper is structured as follows. Section 2 lists similar activity recognition applications and discusses the state
124 of the art in activity recognition with wearable sensors. In Section 3, we present our algorithm to detect and classify
125 goalkeeper training exercises using the signal produced by an inertial sensor. Section 4 presents the results of our
126 evaluation to quantify the performance of our detection and classification algorithms. In Section 5, we discuss the
127 results we obtained and in Section 6 we summarize our contribution and present future research directions towards the
128 realization of a virtual coach for soccer goalkeepers.
129
130

131 2 RELATED WORK 132

133 Activity recognition using wearable sensors has been a field of study for over two decades. In this period, a variety of
134 activity recognition applications have been developed that extract information about the user or her context using
135 sensor signals. This is done for several possible reasons: 1) to assess the performance and correctness of a physical
136 exercise of a patient or athlete, 2) to monitor a physiological parameter or activity over time, 3) to provide feedback to a
137 user about her actions and 4) to adapt a user interface to the user’s context. This section first provides a list of similar
138 activity recognition applications and then discusses the state of the art in recognition methods.
139
140

141 2.1 Activity Recognition Applications 142

143 Activity recognition applications developed so far can be grouped under different fields:
144

- 145 • **Daily activity monitoring.** One of the first activity recognition applications proposed by the research commu-
146 nity used inertial sensors to recognize daily activities (e.g. walking, jogging, standing, sitting) [Bao and Intille
147 2004; Tapia et al. 2004]. Since then, several applications have been studied including the automatic detection
148 of falls [Abbate et al. 2012; Chen et al. 2006] and the recognition and monitoring of dietary activities such as
149 drinking [Amft et al. 2005; Schiboni and Amft 2018] and chewing [Amft 2010; Amft et al. 2005].
- 150 • **Medicine.** Several studies have developed methods to segment strides and extract information from human
151 gait [Barth et al. 2015]. In particular, the gait of patients of Parkinson’s disease has been widely investigated by
152 different research groups [Mariani et al. 2013; Patel et al. 2009, 2006]. Activity recognition applications have also
153
154
155
156

157 been developed to extract objective performance metrics to support clinical assessment during rehabilitation and
158 physical therapy. Performance metrics include the angle of flexion of a leg and the stability while performing a
159 squat during rehabilitation after a knee surgery [Haladjian et al. 2018a, 2015].

- 160 • **Sports.** Wearable activity recognition applications in sports have been developed to detect and classify strokes
161 (i.e. serves) in table tennis [Blank et al. 2015] and in tennis [Yang et al. 2017], determine the part of the shoe
162 used to kick a ball in soccer [Zhou et al. 2016], classify tricks in skateboarding [Groh et al. 2017a], compute
163 the velocity and the length of a jump in ski jumping [Groh et al. 2017b], calculate performance parameters in
164 swimming such as the amount of strokes and time needed per lane [Bächlin et al. 2009] and recognize batting
165 shots in cricket [Khan et al. 2017].
- 166 • **Animal welfare and sports.** Although the field is often referred to as *Human Activity Recognition* (HAR)
167 [Bulling et al. 2014], several activity recognition applications have been developed for non-human users. These
168 include wearable sensors to: detect deviations in the usual gait of cows in order to warn veterinarians about
169 possible lameness-related diseases [Haladjian et al. 2018b, 2017], assess the performance during horse dressage
170 riding [Thompson et al. 2015], recognize gaits and compute the duration of jumps in equestrian show jumping
171 [Echterhoff et al. 2018a,b] and track the activities (i.e. eating, walking, resting) of sheep [Walton et al. 2018].

172 Few research works have studied the use of wearable sensors in soccer applications. Zhou et al. [Zhou et al. 2016]
173 attached a pressure sensitive smart textile matrix to a soccer shoe and studied how to compute the angle at which a
174 soccer ball is kicked. Similarly, Weizman et al. [Weizman and Fuss 2015] developed a smart shoe system with a matrix
175 of pressure sensors and a user interface that displays the force and center of pressure of soccer kicks. Hossain et al.
176 [Hossain et al. 2017] studied the use of wrist-worn sensors to classify motion performed by soccer field players such as
177 passes, kicks, sprints, runs and dribblings. Schuldhaus et al. [Schuldhaus et al. 2016] used a wearable motion sensor in
178 the insole of a soccer shoe to detect specific motions such as dribbling and kicking, which they use to generate video
179 highlights automatically.

180 Other studies have used video cameras to automatically extract metrics from soccer games, such as the positions of
181 the field players and ball trajectories [Figueroa et al. 2006; Müller Junior and Anido 2004]. Our work focuses on tracking
182 soccer goalkeeper training exercises. We do this with an unobtrusive wearable sensor strapped around the glove at the
183 goalkeeper's main hand that does not require calibration or setup.

192 2.2 Activity Recognition Methods

193 Most activity recognition applications developed so far are hand-crafted chains of computations, known as the activity
194 recognition chain [Bulling et al. 2014] to extract information from sensor signals. Multiple methods to segment sensor
195 signals, extract features and to select and prioritize features have been studied in the past. Methods used to recognize
196 patterns in sensor signals proposed so far include template-based methods such as string matching [Stiefmeier et al.
197 2007] and Dynamic Time Warping (DTW) [Barth et al. 2015; Muscillo et al. 2007; Plouffe and Cretu 2015; Seto et al.
198 2015], probabilistic methods such as the Hidden Markov Model (HMM) [Li et al. 2015; Martindale et al. 2017; Schiboni
199 and Amft 2018] and machine learning classifiers such as Support Vector Machines (SVMs) [Haladjian et al. 2018b;
200 Reyes-Ortiz et al. 2016]. Our exercise detection algorithm is based on a memory-efficient peak detection algorithm and
201 a two-step recognition method. In a first step, our recognition method filters out most irrelevant motion patterns that
202 correspond to passes (i.e. when goalkeepers pass the ball back to their trainer) by matching the signal to a pre-computed
203 template using Dynamic Time Warping. Second, it uses a machine learning classifier to determine the actual exercise.
204
205
206
207
208

In the last few years, the community has started investigating the so-called end-to-end methods for activity recognition. These methods extract information directly from the raw data, thus, relieving developers from the tedious work to study data and manually implement and assess an activity recognition algorithm. Different Convolutional Neural Network (CNN) architectures have already surpassed hand-crafted methods in recognition performance [Ha and Choi 2016; Jiang and Yin 2015; Yang et al. 2015; Zeng et al. 2014]. Most notably, Recurrent Neural Networks (RNN) - mostly Long-Short-Term-Memory (LSTM) - have been adapted to exploit the time-dependency of the sensor signals, attaining the highest recognition results so far on large public benchmark datasets [Li et al. 2018; Murahari and Plötz 2018; Neverova et al. 2016; Ordóñez and Roggen 2016; Zeng et al. 2018]. We use a lightweight unobtrusive device that can be worn at the wrist by goalkeepers during a training session for up to 3 hours, or ideally longer. To satisfy these requirements, we designed our system based on a lightweight embedded device that performs a low-cost hand-crafted recognition algorithm locally.

3 METHODS

Figure 2 shows an overview of our recognition system. In this section, we describe the hardware device we used, how we collected the data and the computations we performed to detect and classify goalkeeper training exercises.

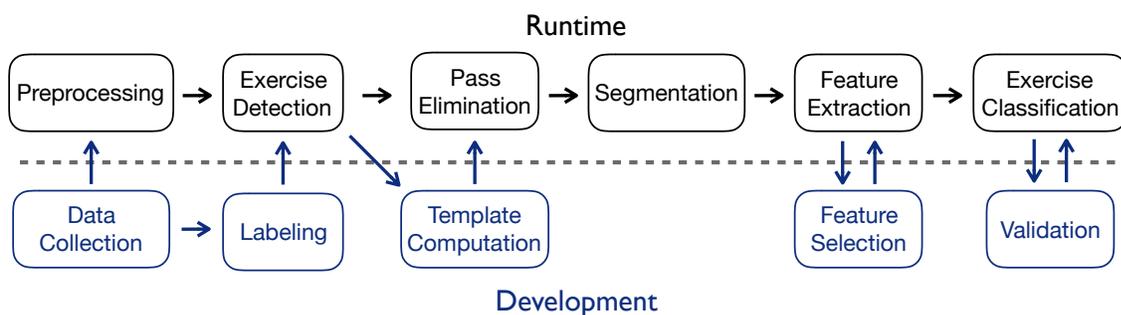


Fig. 2. Overview of the computations we perform to detect and classify goalkeeper training exercises. We separate between computations done at development and runtime.

3.1 Hardware

We collected data using a sensor device called *MicroHub* developed by the company InteractiveWear¹. The *MicroHub* is a modular sensing device developed for wearable applications with an ARM Cortex-M0 microcontroller, Invensense's ICM20602 6-axis (accelerometer and gyroscope) Inertial Measurement Unit (IMU), a Bluetooth Low Energy (BLE) module and an SD card. The accelerometer measures acceleration forces in units of gravity g . Such forces may be static (e.g. the gravity) or dynamic (e.g. motion). The gyroscope measures angular velocity (i.e. the speed of rotation) in degrees per second (dps). The accelerometer and gyroscope were set to their maximum ranges: $\pm 16 g$ and $\pm 2000 dps$, respectively. The data produced by both sensors was recorded with a 16 bit resolution and stored in an SD card mini with a capacity of 2 GB. This made it possible for us to store several hours of training at a sampling rate of 100 Hz without data loss or disconnections, as might have been the case with a wireless communication technology.

¹<http://www.interactive-wear.com/>

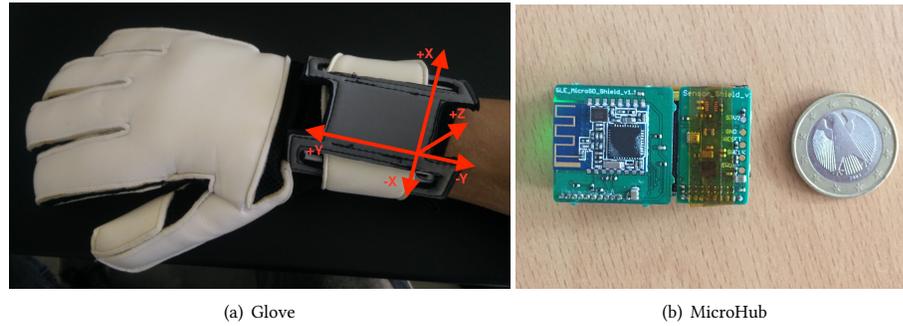


Fig. 3. Smart Glove components and orientation of the accelerometer axes. The x-axis represents side movements, the y-axis forward and backward movements and the z-axis up and down movements.

The Microhub was powered with a 200 mAh Li-Po battery. With this battery, the device remains functional for at least 6 hours while recording raw data continuously and storing it on the SD card. The device's dimensions (including the battery) are: 3.8 x 1.1 x 2.4 cm. To fit the device into a goalkeeper's glove, we designed a mount that is strapped tightly using the glove's strap above the wrist. Since our mount does not require modifications to the glove, it can be strapped around most goalkeeper gloves in the market. The MicroHub is oriented such that the x-axis represents side movements (i.e. to the left and right of the user's hand), the y-axis forward and backward movements (i.e. towards or against the fingers) and the z-axis up and down movements (assuming the palm of the hand is facing the ground). The glove, mount and sensor coordinate system are shown in Figure 3.

Table 1. Data collection plan followed with each goalkeeper. Dive Stand is a variation of a dive where the goalkeeper had to stand up quickly after the dive.

#	Exercise Type	Repetitions
1	Dive	x5 low left, x5 low right, x5 high left, x5 high right
2	Catch	x10 hand catch, x10 body catch and x10 ground catch
3	Dive Stand	x5 low left, x5 low right, x5 high left, x5 high right
4	Throw	x10 high, x10 low
5	Jump Catch	x8

3.2 Data Collection

We collected data from 14 goalkeeper trainees during their training with a professional goalkeeper trainer with over 30 years of experience. During the training, goalkeepers performed different variations of dives, throws and catches. The trainer threw balls at the goalkeepers from different angles and at different intensities. After performing the motion, goalkeepers passed the ball back at the trainer to signal that they were ready for the next repetition. Figure 1 shows goalkeepers executing these exercises during their regular training. The goalkeeper trainees we recorded were male, right handed, 10 to 17 years old, had 1 to 7 years of experience training as goalkeepers and were 1.5 to 2.01 meters tall. Goalkeepers first warmed up, stretched and then started the exercise execution. Table 1 summarizes the exercises

313 performed by goalkeepers and the approximate amount of repetitions they performed. Each training session lasted on
314 average 33 minutes, which led to a file size of approximately 3 MB per recording session.
315

317 3.3 Labeling

318 Each training session lasted approximately one hour. We video-recorded the entire training and annotated every
319 repetition of the training exercises (e.g. dives, catches, throws) as well as other motions goalkeepers performed during
320 the training, such as passes (i.e., when the goalkeeper passed the ball back to the trainer), sprinting, clapping with the
321 hands and bouncing the ball. To synchronize the sensor signal and video recording, we asked goalkeepers to clap three
322 times in the beginning, middle and at the end of the training session. The annotations were done by five individuals and
323 were reviewed by one of the authors to ensure consistency. In total, we labeled 2562 motion instances (1518 training
324 exercises and 1044 instances of other motion, such as passes and hand claps).
325
326
327

329 3.4 Exercise Detection

330 The *Exercise Detection* identifies possible exercises in an incoming stream of motion data. An overview of the different
331 computations our detection algorithm performs is shown in Figure 4. We start by computing the squared magnitude of
332 acceleration along the x, y and z-axes according to:
333

$$334 \text{Magnitude}^2 = a_x^2 + a_y^2 + a_z^2 \quad (1)$$

335 Most exercises cause a peak in the squared magnitude signal due to an impact with the ball or ground. These peaks
336 occur suddenly (i.e., have a duration of 50 ms or less) and hence contain mostly motion in high frequency bands.
337 Therefore, we filter out low-frequency motion using a first order Butterworth high-pass filter with a cutoff frequency
338 $f_c = 25 \text{ Hz}$.
339

340 In the next step, we detect peaks in the filtered squared magnitude signal with a peak detection algorithm that
341 executes for each new value. A new value becomes the peak candidate if it is above a *minimum peak threshold* η and if
342 it is larger than the current peak candidate (in case the peak candidate is already set). Peak candidates become a peak
343 after a *minimum amount of samples* δ elapse. We selected the parameter values $\eta = 8.2 \times 10^6 g^2$ and $\delta = 97 \text{ samples}$
344 because they yielded the best detection performance, as we discuss in the Results section.
345
346
347
348
349

350 3.5 Pass Elimination

351 The peak detection method described in the previous subsection detects movements with a high intensity. A particular
352 movement with high intensity performed frequently by goalkeepers is the pass, as they pass the ball back to the trainers
353 after most exercise repetitions. Since passes are not an exercise that is of interest to goalkeepers, they need to be filtered
354 out. While passes could be filtered out during the classification stage (i.e. by including them as a class in the machine
355 learning classifier), doing so would require extracting features and performing a classification for each detected pass.
356 The goal of the *Pass Elimination* procedure is to eliminate pass instances with a simple heuristic to avoid the more
357 computationally expensive feature extraction and classification computations.
358

359 Most goalkeepers perform passes in a similar way: they sequentially swing the arm back to gain momentum and
360 then forward and release the ball. This motion usually lasts approximately one second. Based on this observation, we
361 designed a method to detect and filter out passes that does not require extracting features or running a prediction for
362
363
364

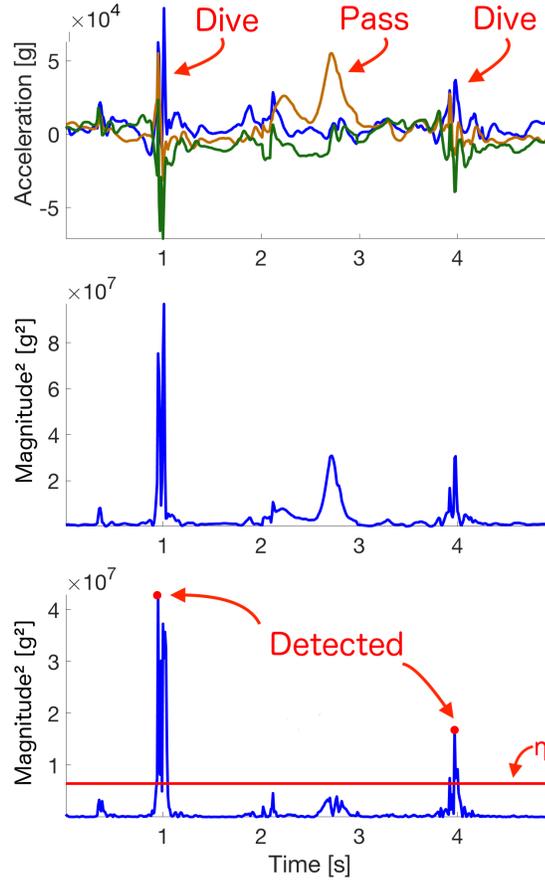


Fig. 4. Sequence of computations performed by our approach to detect exercises. Top: raw accelerometer signal for two dives and a pass. Middle: squared magnitude of the accelerometer signal. Bottom: high-pass filtered squared magnitude of accelerometer signal. Both dives are detected using a peak detector with a minimum peak height $\eta = 8.2 \times 10^6$.

each detected pass. It should be noted that we still include the pass in the different classifiers we studied under the *Other* class, as described in the Classification subsection.

Comparing two pass segments by accumulating the distances of each of their samples sequentially would not account for the fact that passes can vary in speed. As a consequence, this naive comparison procedure would compute a high distance for two similar passes whenever one of the passes was performed at a different speed. Dynamic Time Warping (DTW) is an algorithm that relies in dynamic programming to measure the similarity of two time series that might vary in speed. In particular, it finds the matching between samples in a segment to samples in another segment that lead to a minimal distance between segments.

Previous work in human activity recognition has successfully used DTW to identify human strides by first detecting strides with a peak detector and then comparing them to a stride template [Derawi et al. 2010] or by directly comparing consecutive segments to a stride template [Barth et al. 2015]. Similarly, our method compares the signal around a

417 detected peak to a pass template and discards the detected peaks if the segment around them is similar enough to the
 418 template. Our procedure first creates a segment around the detected peak with index p according to:
 419

$$420 \quad sk_{pass} = [p - a', \dots, p + b'] \quad (2)$$

421
 422 We determined the offsets $a' = 70$ and $b' = 25$ empirically based on our observation that most peaks corresponding
 423 to passes occur approximately 70 samples after the beginning of a pass and that the motion corresponding to a pass
 424 lasts for approximately 95 samples. These segments contain only the acceleration along the y-axis, as we found this
 425 axis to represent best the swing performed during a pass. The DTW algorithm computes the distance between a pass
 426 segment and the pass template based on a cost metric to compare two samples. As a cost metric, we used the absolute
 427 difference in y-acceleration between the two samples.
 428
 429

430 Our procedure discards events if the segments around them have a DTW-distance to the pass template smaller than
 431 a threshold τ . We compared the performance of different values of τ (see section *Results*) and set $\tau = 39000$. The next
 432 subsection describes how we computed the pass template.
 433
 434

435 3.6 Template Computation

436 An ideal pass template S_T has a small distance to as many pass segments S_i and a large distance to as many non-pass
 437 segments as possible. Our template computation procedure selects as the template the pass segment with the minimal
 438 distance to the greatest number of other pass segments. We compute the template in three steps. The first step computes
 439 the distance between every pair of pass segments. To this end, we fill the matrix $M(i, j) \in R^{n \times n}$ with the DTW-distances
 440 of each pass segment S_i to each other pass segment S_j according to:
 441
 442

$$443 \quad M(i, j) = \begin{cases} DTW(S_i, S_j), i, j \in [1, \dots, n], & \text{if } i \neq j \\ \infty, & \text{if } i = j \end{cases} \quad (3)$$

444 where n is the number of pass segments detected by the *Exercise Detection* stage. The distance between a segment to
 445 itself is set to ∞ because the second step counts how many *other* segments a segment is closest in distance to. Thus,
 446 setting $M(i, i) = \infty$ avoids counting a segment as having the closest distance to itself.
 447

448 In the second step, we determine which segments are closest in distance to each other segment. To this end, we define
 449 the matrix $B \in R^{n \times n}$ that contains a 1 at each position $B(i, j)$ if the segment S_i is closest in distance to the segment S_j ,
 450 or a 0 otherwise. In other words, we set a 1 at $B(i, j)$ if $M(i, j)$ is the minimum of the j column in M :
 451
 452

$$453 \quad B(i, j) = \begin{cases} 1, & \text{if } M(i, j) = \min(M(k, j)), \forall k \in [1, \dots, n] \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

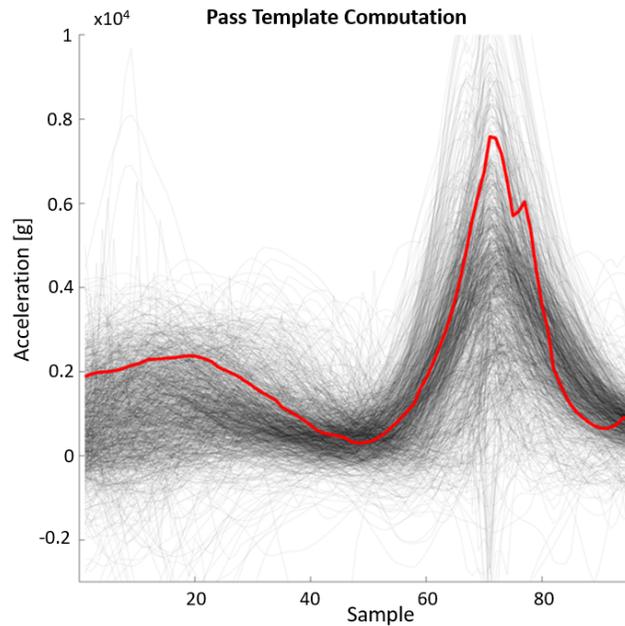
454 The third step counts how many segments a specific segment has minimal distance to. This is done by accumulating
 455 the values in each row of B into a vector C :
 456
 457

$$458 \quad C(i) = \sum_{j=1}^n B(i, j), \forall i \in [1, \dots, n] \quad (5)$$

459 Finally, we perform a voting method to decide which pass segment becomes the template. A naive method to select a
 460 pass template would let every segment *vote* for another pass segment (e.g. the pass segment with shortest distance) and
 461 select the pass segment with largest number of votes. However, this voting method might select an outlier pass segment
 462
 463
 464
 465
 466
 467
 468

469 as a template which is closest in distance to only a few other pass segments. In particular, this might happen if the pass
 470 segments are spread such that each of them are closest to different pass segment candidates without a clear *winner*.
 471

472 Instead, we use the *Instant-runoff* voting method. In computational social choice theory, Instant-runoff is a method
 473 to select a single winner in an election with more than two candidates. The method lets voters rank every candidate in
 474 order of preference. If a candidate obtains the majority of the votes, it is elected as the winner. Otherwise, the candidate
 475 with least amount of votes is eliminated and the election is repeated with the remaining candidates. In our adapted
 476 version of the method, every pass segment is a voter and a candidate to become the template at the same time and pass
 477 templates are ranked based on the number of other segments they are closest to in DTW-distance. In each iteration, our
 478 algorithm eliminates the segment S_m with smallest distance to the *least* other segments: $\{m \mid C_m = \min(C_i), \forall i = 1 \dots n\}$
 479 from the list of candidates. This procedure is repeated $n - 2$ times. In each iteration, the matrix B and vector C are
 480 recomputed. After $n - 2$ iterations, only one segment is left, which is selected as template. Figure 5 shows the template
 481 selected with this procedure on top of every other pass segment in our data set.
 482
 483



507 Fig. 5. Acceleration along the y-axis of every pass segment (black) and the template (red).

511 3.7 Segmentation

512 For each event that was not eliminated in the *Pass Elimination* procedure, we create a segment range k around the peak
 513 location x_p according to:
 514

$$516 k = [x_{p-a}, \dots, x_{p+b}] \quad (6)$$

517 We set $a = 130$ and $b = 90$ empirically based on a visual comparison of the motion produced by the different exercise
 518 instances, as shown in Figure 6. The *Segmentation* stage produces a segment $S(k)$ containing the acceleration, angular
 519

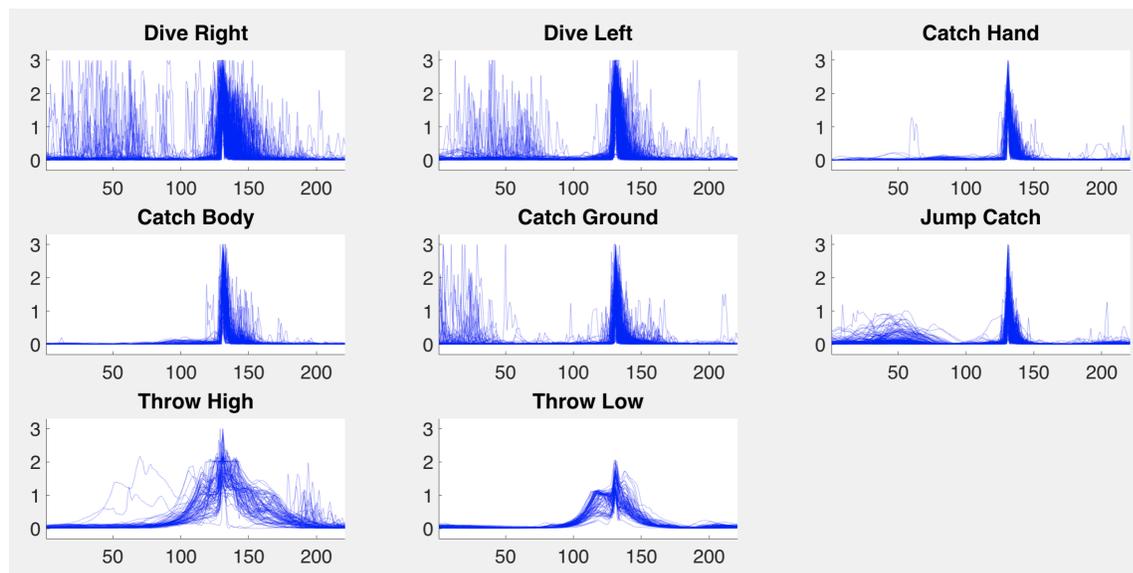


Fig. 6. Squared magnitude of acceleration of every segment produced by the segmentation algorithm plotted on top of each other and grouped by exercise.

velocity and squared magnitude of each sample in the range k . Figure 6 displays the segments produced by our algorithm after this step, which are used for feature extraction, as described in the next section.

3.8 Feature Extraction

We studied a set of statistical and heuristic features. The statistical features we chose have been previously used in similar wearable activity recognition applications [Blank et al. 2015; Bulling et al. 2014; Haladjian et al. 2017]. The list of features we studied is summarized in Table 2. The statistical features are computed on all three axes of acceleration and angular velocity. Heuristic features are computed on all three axes of the acceleration vector, all three axes of angular velocity and on the squared magnitude of acceleration computed with Equation 1.

We designed heuristic features to highlight the differences in motion across the different exercise variations. The peaks detected in the *Exercise Detection* are located at the moment of maximum intensity of acceleration (e.g., contact with the ground, contact with the ball, release of a ball). The different exercises have a different acceleration before the peak and deceleration after the peak along each axis. To capture the differences in acceleration and deceleration, we split the segment in three parts based on the position of the peak p as: $k_{left} = [1, \dots, p - c]$, $k_{center} = [p - c, \dots, p + c]$ and $k_{right} = [p + c, \dots, n]$ where n is the number of samples in the segment and c is a constant. We set $c = 20$ by observing the signals grouped by exercise, as shown in Figure 6. Figure 7 shows the mean acceleration along the z -axis of all three parts computed from every right and left dive in our data set.

Some classifiers calculate the distance between feature vectors. Features with a large range of values might contribute more to the distance than features with smaller ranges of values. Therefore, we normalize every feature f_i in the segment to have zero mean and a standard deviation in the range $[-1, 1]$ with:

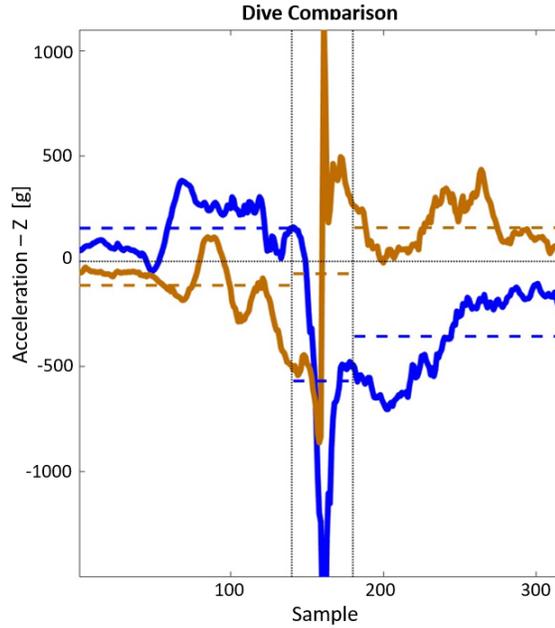


Fig. 7. Dive right (blue) and left (orange) computed by averaging the acceleration along the z-axis of every dive segment in our data set. Most right dives have a positive acceleration mean before the peak and negative during and after the peak. Most left dives have the opposite acceleration.

Table 2. List of features we studied. Statistical features are computed on the entire segment. Heuristic features are computed on each segment part: k_{left} , k_{middle} and k_{right} .

Type	Features
Statistical	mean, median, standard deviation (std), accumulated squared magnitude of acceleration, skewness, kurtosis, zero crossings (zcr), peak to peak amplitude (p2p), root mean square (rms), minimum value (min), maximum value (max), correlation between acceleration axes (corr)
Heuristic	mean, standard deviation (std), zero crossing, quantile, peaks

$$\bar{f}_i = \frac{f_i - \mu(f_i)}{\sigma(f_i)} \quad (7)$$

where $\mu(f_i)$ and $\sigma(f_i)$ are the mean and standard deviation of the feature f_i calculated based on every segment in our data set.

3.9 Feature Selection

Extracting and classifying a larger number of features leads to a larger number of computations that have to be done on the wearable device and make machine learning classifiers more prone to overfitting. For this reason, we studied how to reduce the number of features with a feature selection algorithm called *minimum Redundancy Maximum Relevance* (mRMR) [Peng et al. 2005]. The 20 most relevant features selected by mRMR and the signals they are computed on are listed in Table 3.

Table 3. Features selected for classification using mRMR.

Feature	Computed on
Quantile	Az_3, Az_2, Az_4, Ay_2
Amount of Peaks	Squared Magnitude on k_{right}
Cross-correlation	$Ay - Az$
Mean (Segment)	Ay, Gz
Maximum	Gx, Ay
Root Mean Square	Gz, Gx
Standard Deviation	Ay, Az
Median Absolute Deviation	Ax, Gx
Median	Az
Mean	Az on k_{right} , Ay on k_{right} , Az on k_{middle}

3.10 Exercise Classification

We compared the performance of different classifiers: Naive Bayes, Decision Tree, Random Forest, Support Vector Machine (SVM) with linear and Radial Based Function (RBF) kernels, k-nearest neighbors (kNN) and a Neural Network. For each classifier, we optimized the performance of the following parameters:

- **SVM (RBF and linear kernel):** cost parameter $c \in \{1, 2, 3, 5, 8, 13, 21, 34, 55\}$, kernel coefficient $\gamma \in \{0.0, 0.1, 0.2, \dots, 10.0\}$, tolerance for stopping criterion $tol \in \{0.001, 0.002, 0.003, \dots, 0.3\}$.
- **Decision Tree:** maximum depth $max_depth \in \{1, 2, 3, \dots, 30\}$.
- **Random Forest:** maximum depth $max_depth \in \{1, 2, 3, \dots, 30\}$, number of features to consider when looking for the best split $max_features \in \{1, 2, 3, \dots, 11\}$, minimum number of samples required to reach a leaf node $min_samples_leaf \in \{1, 2, 3, \dots, 11\}$, minimum number of samples required to split an internal node $min_samples_split \in \{2, 3, 4, \dots, 11\}$.
- **kNN:** $k \in \{1, 2, 3, \dots, 15\}$.
- **Neural Network:** number of hidden layers $n \in \{1, 2\}$, size of hidden layers $hidden_layer_sizes \in \{10, 20, \dots, 3000\}$, solver for weight optimization $solver \in \{lbfgs, adam\}$, tolerance for the optimization $tol \in \{10^{-2}, 10^{-3}, \dots, 10^{-6}\}$.

3.11 Validation

To avoid evaluating the performance of the different methods we presented with the same data used to optimize the parameters η (minimum peak height), δ (minimum amount of samples between peaks), τ (DTW distance threshold) and the hyperparameters of our machine learning classifiers, we used a nested cross-validation procedure. In an outer loop, we tested different values for the parameters and the inner loop performed a leave-one-subject-out cross-validation. First, we found the optimal values for the parameters η and δ with a grid search in the ranges $[3 \times 10^6, 18 \times 10^6]$ and $[10, 200]$ at intervals of 10^5 and 1, respectively. Then, we optimized τ using a linear search in the range $[1 \times 10^2 - 2 \times 10^5]$ at intervals of 5×10^3 . Finally, we optimized the hyperparameters of the classifiers with a randomized search followed by a grid search. The randomized search gave us optimal value ranges for each parameter and the grid search found the optimal values within each range. In each iteration of the inner loop, a different goalkeeper data file was excluded from the training set and used for assessing the performance of our methods. We selected the parameters that lead to the best average performance across player files.

4 RESULTS

Our algorithm first detects exercise instances from a stream of sensor values, and then classifies those detected events. Applications that rely on our algorithm would only use the final classifications. However, to provide insight about the different stages of our recognition pipeline, in this section we first report on the performance of the exercise detection method, and then discuss the classification method.

4.1 Exercise Detection

This subsection reports on the performance results for the computations we presented in the *Preprocessing*, *Exercise Detection* and *Pass Elimination* subsections. The detection of exercises can be understood as a binary classification (i.e. a segment is either classified as relevant or not relevant). To validate our exercise detection procedure, we compared the exercise labels to the detected segments as follows:

True Positive (TP). A detected segment that included a relevant exercise label.

False Positive (FP). A detected segment that did not include any relevant exercise label.

False Negative (FN). An exercise labeled as relevant that was not included in any of the detected segments.

Figure 8 shows a performance comparison of η and δ and Figure 9 presents the F-Score for different values of τ . Our detection method achieves an F-Score of 92.2%. In total, 1424 of the 1518 exercise repetitions were detected by our detection method (Recall = 93.8 %) whereas 148 instances of irrelevant movements were also detected as exercises (Precision = 90.6%).

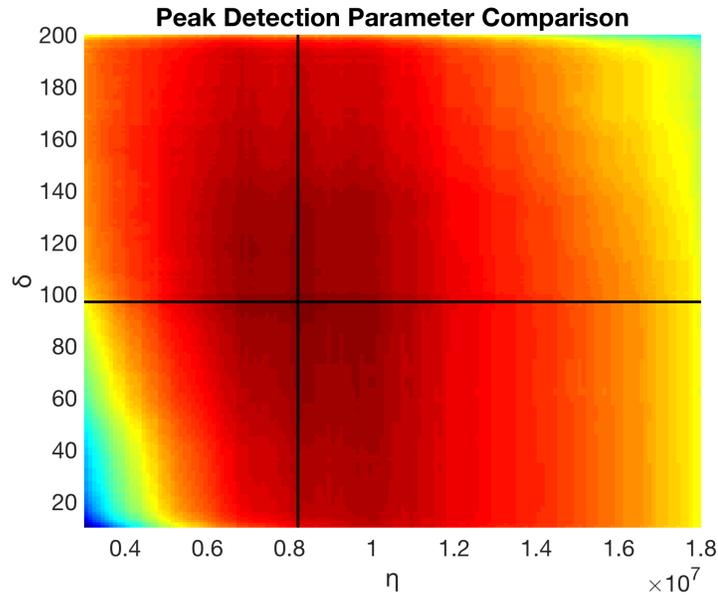


Fig. 8. Comparison of the F-Score for different values of η and δ . Red values indicate higher F-Scores. Line markers were placed at the values leading to the highest F-Score ($\eta = 8.2 \times 10^6$ and $\delta = 97$).

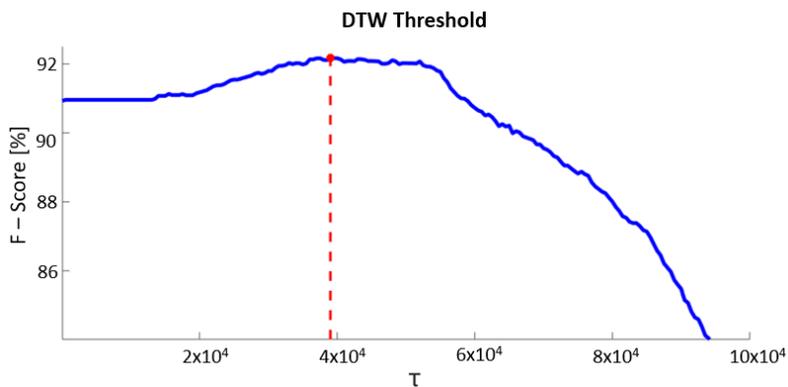


Fig. 9. F-Score of the exercise detection for different values of τ . $\tau = 39000$ achieves the highest F-Score: 92.2% (Precision 90.6%, Recall 93.8%) with the peak detection parameters: $\eta = 8.2 \times 10^6$ and $\delta = 97$.

The performance of our detection algorithm at detecting the different exercises is shown in Table 4.

Table 4. Amount of instances of each motion type detected by our approach and annotated in our data set.

	Exercise	Detected	Annotated	%
Relevant	Dive Right	285	286	99.7%
	Dive Left	248	251	98.8%
	Catch Hand	144	146	98.6%
	Catch Body	212	217	97.7%
	Catch Ground	145	158	91.8%
	Jump Catch	112	150	74.7%
	Throw High	141	145	97.2%
	Throw Low	137	165	83.0%
	Other	148	1044	14.2%

4.2 Exercise Classification

The segments detected as relevant by our exercise detection method are classified into the different exercises. Table 5 shows the accuracy of the different classification algorithms we tested. The classifier that achieved the highest accuracy is the SVM with an RBF Kernel and a box constraint $c = 7$ (accuracy: 96.5%). The confusion matrix and classification performance per exercise computed with this classifier are shown in Tables 6 and 7. The kNN classifier was tested with $k = 8$, which led to the maximum accuracy among the different variations of the kNN algorithm of 92.2%.

Table 5. Performance of different classifiers at classifying goalkeeper training exercises. SVM performed best with an RBF kernel and parameters: $c = 7$, $\gamma = 0.0095$, $decision_function_shape = ovr$ and $tol = 0.001$

	Accuracy	Precision	Recall
Naive Bayes	74.4%	76.1%	76.8%
Decision Tree	77.9%	78.8%	79.1%
Random Forest	87.2%	88.8%	88.3%
SVM (Linear)	93.0%	94.6%	93.6%
SVM (RBF)	96.5%	96.9%	96.7%
K-Nearest Neighbors	92.1%	92.5%	93.8%
Neural Network	93.8%	93.9%	93.5%

Table 6. Confusion matrix computed using an SVM classifier with an RBF kernel and cost parameter $c = 7$.

		Annotated							
		DR	DL	CH	CB	CG	JC	TH	TL
Predicted	DR	271	12	2	0	0	0	0	0
	DL	11	236	0	0	0	1	0	0
	CH	2	0	141	0	0	1	0	0
	CB	0	0	1	207	3	1	0	0
	CG	1	1	0	8	135	0	0	0
	JC	0	1	2	0	0	108	1	0
	TH	0	0	0	0	0	0	140	1
	TL	0	0	0	0	0	0	1	136

Table 7. Accuracy for each training exercise computed using the SVM classifier with an RBF kernel and cost parameter $c = 7$.

	Precision	Recall	F-Score
Dive Right	95.1%	95.1%	95.1%
Dive Left	94.4%	95.2%	94.8%
Catch Hand	96.6%	97.9%	97.2%
Catch Body	96.3%	97.6%	97.0%
Catch Ground	97.8%	93.1%	95.4%
Jump Catch	97.3%	96.4%	96.9%
Throw High	98.6%	99.3%	98.9%
Throw Low	99.3%	99.3%	99.3%

5 DISCUSSION

The results we presented indicate that it is possible to detect and classify goalkeeper training exercises accurately. Our detection approach detects 93.8% of the relevant exercise instances with a precision of 90.6% and the SVM classifier with an RBF kernel achieves a classification accuracy of 96.5%.

We focused our analysis on the most common variations of the goalkeeper training exercises performed during a training session. In a free-training (i.e. soccer match), the amount of relevant exercises might be considerably sparser and goalkeepers might perform other exercises such as sprints, kicking the ball with the leg or warm up exercises which we did not consider in this study. Therefore, we expect a lower precision in a free-training than the one we presented in this study. In the future, the wider variability of a free-training should be studied to identify new classes and train a classifier accordingly.

833 Our detection approach detected over 97% of the instances of most exercise types and filtered out most of the
834 irrelevant motions. Only 148 instances of irrelevant motion were wrongly detected as relevant exercises, which is a
835 small fraction (14.2%) of the total amount of instances of irrelevant motion we annotated (1044) and an even smaller
836 fraction of every irrelevant motion goalkeepers performed during the training session which we did not annotate (e.g.,
837 walking, running, picking up a balls from the ground, receiving balls passed at them without any specific technique,
838 etc.). On the other hand, only 74.2% of the jump catches and 83.0% of the low throws were detected. The reason is that
839 these exercises contain low frequency motion that is attenuated by the high-pass filter, which causes the peak detector
840 to miss the peaks produced by some of these exercise instances. These exercises could be detected by lowering the
841 peak threshold η , but this would also increase the false positive detection rate. The SVM classifier with an RBF kernel
842 produced the highest accuracy of 96.5%.

843 Figure 9 shows that without our *Pass Elimination* procedure ($\tau = 0$), the detection procedure achieves an F-Score
844 of 90.9% and that our *Pass Elimination* procedure improves this performance to 92.2% when $\tau = 39000$. Besides the
845 improvement in recognition performance, an important benefit of this procedure is that every time a pass is eliminated,
846 no features have to be extracted, stored in flash memory (or transmitted wirelessly) and classified, resulting in a lower
847 energy consumption. The procedure can detect and eliminate more pass instances by increasing the template distance
848 threshold τ . However, increasing τ over a value of 39000 leads to the elimination of more relevant exercise instances
849 than irrelevant ones. The main reason for this is that low throws have a similar signature to passes, causing low throws
850 executed by players with a low intensity to be incorrectly matched to the pass template. Nevertheless, the procedure
851 we described can be used as-is in other applications if a particular class has a consistent signature that is different
852 enough from other classes. Furthermore, future work could study different template selection strategies. In particular,
853 we have selected our template based on its similarity to other instances of the same class. In the future, the distances to
854 other classes could also be used as part of the optimization metric to select a template. Furthermore, additional signals
855 (besides only the y-axis of the accelerometer vector) could be used to compare segments to the template and different
856 amounts of templates could be selected per class. Finally, future work could study the recognition and computational
857 performance when using the distances to different templates as features within a machine learning classifier.

858 A possible threat to validity to our evaluation procedure is that we haven't used a test split to assess the performance
859 of the *Pass Elimination* method. As we assessed the performance of the *Pass Elimination* method with the same data we
860 used to select the pass template, the performance we presented might be slightly higher than on unseen data. In the
861 future, the collection of an additional data set will make it possible to select a pass template and assess the performance
862 of a *Pass Elimination* method with different data.

863 In order to enable further data analysis and to improve our recognition algorithms by means of online machine
864 learning, it would be convenient to have access to the feature vectors produced by our algorithm outside of the wearable
865 device. Since streaming feature vectors would require a mobile phone nearby during a training session, we decided to
866 store the feature vectors on the wearable device and transmit them to a mobile device after the training over Bluetooth
867 Low Energy. Since we set a minimal distance between peaks of 97 samples (0,97 seconds), at most 3711 motion instances
868 (relevant or irrelevant) can be detected by our approach during an hour of training. The features and a timestamp for
869 every detected motion instance could be stored in less than 304 kB memory (3711 instances x 21 features and timestamp
870 x 4 bytes per feature and timestamp). Furthermore, our detection algorithm detected a total of 1572 motion instances in
871 our data set (computed by aggregating the values in the *Detected* column in Table 4). As our data set consists of 14
872 training sessions, our algorithm would detect an average of 112 motion instances per training session, which could be
873 stored in approximately 9,6 kB memory (112 instances x 21 features and timestamp represented with 4 bytes each).

6 CONCLUSIONS

We have described a method to detect and classify goalkeeper training exercises. Detecting and classifying these exercises represents a first step towards a virtual coach that gives goalkeepers relevant and objective feedback about their exercise executions. A major challenge we addressed in this paper was on detecting the relevant training exercises while avoiding other movements goalkeepers perform during a training session. We noticed that specially younger goalkeepers tend to perform hectic movements that lead to a larger amount of false positive detections. We showed that a careful preprocessing of the data using a high-pass filter in combination with Dynamic Time Warping can be used to reduce the number of false positive detections. Furthermore, we achieved an exercise detection recall of 93.8% (precision: 90.6%) and a classification accuracy of 96.5%. Our segmentation approach and the methods we used to extract heuristic features and automatically select the most relevant ones can be reused in other applications that require detecting or classifying specific events such as strokes or shots in ball-based sports or strides in gait analysis applications.

Our work is by no means finished. An immediate next step is to investigate what performance metrics are relevant to goalkeepers and which of those can be extracted accurately with our smart glove. Based on our discussions with coaches and experienced goalkeepers, relevant performance metrics include: *'the area of the goal where most balls are missed or caught'*, *'the time needed to stand up on the feet after a dive'* and *'the maximum height and length achieved during a dive'*. Future work should study how to compute different performance metrics with the segments detected and classified by our algorithm.

Before goalkeepers are able to benefit from the system we propose, their needs as well as possible interaction modalities for a user interface will have to be investigated. Some open questions are: What information are goalkeepers interested in and how should this information be provided to them? Should the system provide goalkeepers live feedback while in the field or should the feedback be given offline (e.g. while in the locker room or at home)? To keep the device lightweight while exploiting the fact that most users nowadays have a smartphone, similar systems offer users an interface over a smartphone. However, goalkeepers don't always take their smartphone to the field and even if they did they would have to take off their gloves to operate it via touch. An alternative would be a voice-based interaction, but a soccer pitch is loud so goalkeepers would have to get close to the phone. Another question that raises is: where could a smartphone be placed (e.g. behind the goal) such that it does not disturb the goalkeeper and cannot be hit by a ball?

Furthermore, a larger data set is likely to improve the accuracy and reveal corner cases. As we collect more data, we might discover new irrelevant motions performed frequently by players (e.g. as we realized previously that some players bounce the ball before a throw). These motions need to be considered by our algorithm (e.g. included in the classifier). Furthermore, future work should add support for left-handed goalkeepers, which we did not consider in this study. It might also be interesting to investigate how our recognition method could adapt itself based on data generated after its deployment to the market (e.g. with online machine learning or with a user-dependent recognition).

We are currently investigating the suitability of a computer vision approach to recognize training exercises and extract performance metrics from them. So far, we successfully used a pre-trained pose estimation algorithm to obtain the body joints (wrists, hips, knees) of the goalkeeper from an image frame. These joints could be combined with a machine learning algorithm to extract different exercise execution metrics. We believe that a computer vision approach has a higher potential to accurately recognize exercise performance metrics relevant to goalkeepers. At the same time, the approach would pose other challenges such as occlusion (e.g. a coach walking in front of the camera), confusion when multiple players are detected in an image frame and would impose a higher degree of involvement from goalkeepers to setup the camera in the field.

REFERENCES

- 937 Stefano Abbate, Marco Avvenuti, Francesco Bonatesta, Guglielmo Cola, Paolo Corsini, and Alessio Vecchio. 2012. A smartphone-based fall detection
938 system. *Pervasive and Mobile Computing* 8, 6 (2012), 883–899.
- 939 Oliver Amft. 2010. A wearable earpad sensor for chewing monitoring. In *SENSORS, 2010 IEEE*. IEEE, 222–227.
- 940 Oliver Amft, Holger Junker, and Gerhard Tröster. 2005. Detection of eating and drinking arm gestures using inertial body-worn sensors. In *Ninth IEEE
941 International Symposium on Wearable Computers (ISWC'05)*. IEEE, 160–163.
- 942 Marc Bächlin, Kilian Förster, and Gerhard Tröster. 2009. SwimMaster: a wearable assistant for swimmer. In *Proceedings of the 11th international conference
943 on Ubiquitous computing*. ACM, 215–224.
- 944 Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing*.
945 Springer, 1–17.
- 946 Jens Barth, Cäcilia Oberndorfer, Cristian Pasluosta, Samuel Schüle, Heiko Gassner, Samuel Reinfelder, Patrick Kugler, Dominik Schuldhuis, Jürgen
947 Winkler, Jochen Klucken, and Others. 2015. Stride segmentation during free walk movements using multi-dimensional subsequence dynamic time
948 warping on inertial sensor data. *Sensors* 15, 3 (2015), 6419–6440.
- 949 Peter Blank, Julian Hoßbach, Dominik Schuldhuis, and Bjoern M Eskofier. 2015. Sensor-based stroke detection and stroke type classification in table
950 tennis. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 93–100.
- 951 Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys
952 (CSUR)* 46, 3 (2014), 33.
- 953 Jay Chen, Karric Kwong, Dennis Chang, Jerry Luk, and Ruzena Bajcsy. 2006. Wearable sensors for reliable fall detection. In *2005 IEEE Engineering in
954 Medicine and Biology 27th Annual Conference*. IEEE, 3551–3554.
- 955 Mohammad O Derawi, Patrick Bours, and Kjetil Holien. 2010. Improved cycle detection for accelerometer based gait authentication. In *2010 Sixth
956 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 312–317.
- 957 Jessica Echterhoff, Juan Haladjian, and Bernd Brügge. 2018a. Gait Analysis in Horse Sports. In *Proceedings of the Fifth International Conference on
958 Animal-Computer Interaction*. ACM, 3.
- 959 Jessica Echterhoff, Juan Haladjian, and Bernd Brügge. 2018b. Gait and Jump Classification in Modern Equestrian Sports. In *Proceedings of the 2018 ACM
960 International Symposium on Wearable Computers*. ACM, 88–91.
- 961 Pascual J Figueroa, Neucimar J Leite, and Ricardo M L Barros. 2006. Tracking soccer players aiming their kinematical motion analysis. *Computer Vision
962 and Image Understanding* 101, 2 (2006), 122–135.
- 963 Benjamin H Groh, Martin Fleckenstein, Thomas Kautz, and Bjoern M Eskofier. 2017a. Classification and visualization of skateboard tricks using wearable
964 sensors. *Pervasive and Mobile Computing* 40 (2017), 42–55.
- 965 Benjamin H Groh, Frank Warschun, Martin Deininger, Thomas Kautz, Christine Martindale, and Bjoern M Eskofier. 2017b. Automated Ski Velocity and
966 Jump Length Determination in Ski Jumping Based on Unobtrusive and Wearable Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and
967 Ubiquitous Technologies* 1, 3 (2017), 53.
- 968 Sojeong Ha and Seungjin Choi. 2016. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors.
969 In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 381–388.
- 970 Juan Haladjian, Katharina Bredies, and Bernd Bruegge. 2018a. KneeHapp Textile: A Smart Textile System for Rehabilitation of Knee Injuries. In *Proceedings
971 of the 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 9–12.
- 972 Juan Haladjian, Johannes Haug, Stefan Nüske, and Bernd Bruegge. 2018b. A Wearable Sensor System for Lameness Detection in Dairy Cattle. *Multimodal
973 Technologies and Interaction* 2, 2 (2018), 27.
- 974 Juan Haladjian, Zardosht Hodaie, Stefan Nüske, and Bernd Brügge. 2017. Gait Anomaly Detection in Dairy Cattle. In *Proceedings of the Fourth International
975 Conference on Animal-Computer Interaction (ACI2017)*. ACM, New York, NY, USA, 8:1–8:8. <https://doi.org/10.1145/3152130.3152135>
- 976 Juan Haladjian, Zardosht Hodaie, Han Xu, Mertcan Yigin, Bernd Bruegge, Markus Fink, and Juergen Hoeher. 2015. KneeHapp: A Bandage for Rehabilitation
977 of Knee Injuries. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM
978 International Symposium on Wearable Computers*. ACM, 181–184.
- 979 H M Sajjad Hossain, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. 2017. SoccerMate: A personal soccer attribute profiler using wearables. In *Pervasive
980 Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 164–169.
- 981 Wenchao Jiang and Zhaozheng Yin. 2015. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of
982 the 23rd ACM international conference on Multimedia*. ACM, 1307–1310.
- 983 Aftab Khan, James Nicholson, and Thomas Plötz. 2017. Activity Recognition for Quality Assessment of Batting Shots in Cricket using a Hierarchical
984 Representation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 62.
- 985 Frédéric Li, Kimiaki Shirahama, Muhammad Nisar, Lukas Köping, and Marcin Grzegorzec. 2018. Comparison of feature learning methods for human
986 activity recognition using wearable sensors. *Sensors* 18, 2 (2018), 679.
- 987 Zhen Li, Zhiqiang Wei, Yaofeng Yue, Hao Wang, Wenyan Jia, Lora E Burke, Thomas Baranowski, and Mingui Sun. 2015. An adaptive hidden markov
988 model for activity recognition based on a wearable multi-sensor device. *Journal of medical systems* 39, 5 (2015), 57.
- Benoit Mariani, Mayté Castro Jiménez, François J G Vingerhoets, and Kamiar Aminian. 2013. On-shoe wearable sensors for gait and turning assessment
of patients with Parkinson’s disease. *IEEE transactions on biomedical engineering* 60, 1 (2013), 155–158.

- 989 C F Martindale, M Strauss, H Gaßner, J List, M Müller, J Klucken, Z Kohl, and B M Eskofier. 2017. Segmentation of gait sequences using inertial sensor
990 data in hereditary spastic paraplegia. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.
991 1266–1269. <https://doi.org/10.1109/EMBC.2017.8037062>
- 992 Bruno Müller Junior and Ricardo de Oliveira Anido. 2004. Distributed real-time soccer tracking. In *Proceedings of the ACM 2nd international workshop on*
993 *Video surveillance & sensor networks*. ACM, 97–103.
- 994 Vishvak S Murahari and Thomas Plötz. 2018. On attention models for human activity recognition. In *Proceedings of the 2018 ACM International Symposium*
995 *on Wearable Computers*. ACM, 100–103.
- 996 Rossana Muscillo, Silvia Conforto, Maurizio Schmid, Paolo Caselli, and Tommaso D'Alessio. 2007. Classification of motor activities through derivative
997 dynamic time warping applied on accelerometer data. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology*
998 *Society*. IEEE, 4930–4933.
- 999 Natalia Neverova, Christian Wolf, Griffin Lacey, Lex Fridman, Deepak Chandra, Brandon Barbello, and Graham Taylor. 2016. Learning human identity
from motion patterns. *IEEE Access* 4 (2016), 1810–1820.
- 1000 Francisco Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition.
1001 *Sensors* 16, 1 (2016), 115.
- 1002 S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato. 2009. Monitoring Motor Fluctuations
1003 in Patients With Parkinson's Disease Using Wearable Sensors. *IEEE Transactions on Information Technology in Biomedicine* 13, 6 (nov 2009),
1004 864–873. <https://doi.org/10.1109/TITB.2009.2033471>
- 1005 Shyamal Patel, Delsey Sherrill, Richard Hughes, Todd Hester, Theresa Lie-Nemeth, Paolo Bonato, David Standaert, and Nancy Huggins. 2006. Analysis of
1006 the Severity of Dyskinesia in Patients with Parkinson's Disease via Wearable Sensors. In *International Workshop on Wearable and Implantable Body*
1007 *Sensor Networks (BSN'06)*. IEEE, 123–126. <https://doi.org/10.1109/BSN.2006.10>
- 1008 Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and
1009 min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.
- 1010 Guillaume Plouffe and Ana-Maria Cretu. 2015. Static and dynamic hand gesture recognition in depth data using dynamic time warping. *IEEE transactions*
1011 *on instrumentation and measurement* 65, 2 (2015), 305–316.
- 1012 Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-aware human activity recognition using smartphones.
Neurocomputing 171 (2016), 754–767.
- 1013 Giovanni Schiboni and Oliver Amft. 2018. Sparse natural gesture spotting in free living to monitor drinking with wrist-worn inertial sensors. In *Proceedings*
1014 *of the 2018 ACM International Symposium on Wearable Computers*. ACM, 140–147.
- 1015 Dominik Schuldhuis, Carolin Jakob, Constantin Zwick, Harald Koerger, and Bjoern M Eskofier. 2016. Your personal movie producer: generating highlight
1016 videos in soccer using wearables. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, 80–83.
- 1017 S Seto, W Zhang, and Y Zhou. 2015. Multivariate Time Series Classification Using Dynamic Time Warping Template Selection for Human Activity
1018 Recognition. In *2015 IEEE Symposium Series on Computational Intelligence*. 1399–1406. <https://doi.org/10.1109/SSCI.2015.199>
- 1019 Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. 2007. Gestures are strings: efficient online gesture spotting and classification using string
1020 matching. In *Proceedings of the ICST 2nd international conference on Body area networks*. ICST (Institute for Computer Sciences, Social-Informatics
1021 and ...), 16.
- 1022 Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *International*
conference on pervasive computing. Springer, 158–175.
- 1023 Robin Thompson, Ilias Kyriazakis, Amey Holden, Patrick Olivier, and Thomas Plötz. 2015. Dancing with horses: automated quality feedback for dressage
1024 riders. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 325–336.
- 1025 Emily Walton, Christy Casey, Jurgen Mitsch, Jorge A Vázquez-Diosdado, Juan Yan, Tania Dottorini, Keith A Ellis, Anthony Winterlich, and Jasmeet Kaler.
1026 2018. Evaluation of sampling frequency, window size and sensor position for classification of sheep behaviour. *Royal Society open science* 5, 2 (2018),
171442.
- 1027 Yehuda Weizman and Franz Konstantin Fuss. 2015. Sensor Array Design and Development of Smart Sensing System for Kick Force Visualization in
1028 Soccer. *Procedia Technology* 20 (2015), 138–143.
- 1029 Disheng Yang, Jian Tang, Yang Huang, Chao Xu, Jinyang Li, Liang Hu, Guobin Shen, Chieh-Jan Mike Liang, and Hengchang Liu. 2017. TennisMaster: an
1030 IMU-based online serve performance evaluation system. In *Proceedings of the 8th Augmented Human International Conference*. ACM, 17.
- 1031 Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. 2015. Deep convolutional neural networks on multichannel time
1032 series for human activity recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- 1033 Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu. 2018. Understanding and improving recurrent
1034 networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*.
1035 ACM, 56–63.
- 1036 Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. Convolutional neural networks for human activity
1037 recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 197–205.
- 1038 Bo Zhou, Harald Koerger, Markus Wirth, Constantin Zwick, Christine Martindale, Heber Cruz, Bjoern Eskofier, and Paul Lukowicz. 2016. Smart soccer
1039 shoe: monitoring foot-ball interaction with shoe integrated textile pressure sensor matrix. In *Proceedings of the 2016 ACM International Symposium on*
1040 *Wearable Computers*. ACM, 64–71.
- 1040 Manuscript submitted to ACM