Frame-Level Event Detection in Athletics Videos with Pose-Based Convolutional Sequence Networks

Moritz Einfalt
moritz.einfalt@informatik.uni-augsburg.de
University of Augsburg, Germany

Charles Dampeyrou
charles.dampeyrou@ensta.fr
ENSTA Paris, France

Dan Zecha
dan.zecha@informatik.uni-augsburg.de
University of Augsburg, Germany

Rainer Lienhart
rainer.lienhart@informatik.uni-augsburg.de
University of Augsburg, Germany

ABSTRACT

In this paper we address the problem of automatic event detection in athlete motion for automated performance analysis in athletics. We specifically consider the detection of stride-, jump- and landing related events from monocular recordings in long and triple jump. Existing work on event detection in sports often uses manually designed features on body and pose configurations of the athlete to infer the occurrence of events. We present a two-step approach, where temporal 2D pose sequences extracted from the videos form the basis for learning an event detection model. We formulate the detection of discrete events as a sequence translation task and propose a convolutional sequence network that can accurately predict the timing of event occurrences. Our best performing architecture achieves a precision/recall of 92.3%/89.0% in detecting start and end of ground contact during the run-up and jump of an athlete at a temporal precision of ±1 frame at 200Hz. The results show that 2D pose sequences are a suitable motion representation for learning event detection in a sequence-to-sequence framework.

CCS CONCEPTS

• **Computing methodologies → Visual content-based indexing and retrieval**; **Neural networks**; Object detection.

KEYWORDS

event detection, video indexing, computer vision in sports, convolutional sequence modeling

ACM Reference Format:
Moritz Einfalt, Charles Dampeyrou, Dan Zecha, and Rainer Lienhart. 2019. Frame-Level Event Detection in Athletics Videos with Pose-Based Convolutional Sequence Networks. In *2nd International Workshop on Multimedia Content Analysis in Sports (MMSports '19), October 25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3347318.3355525

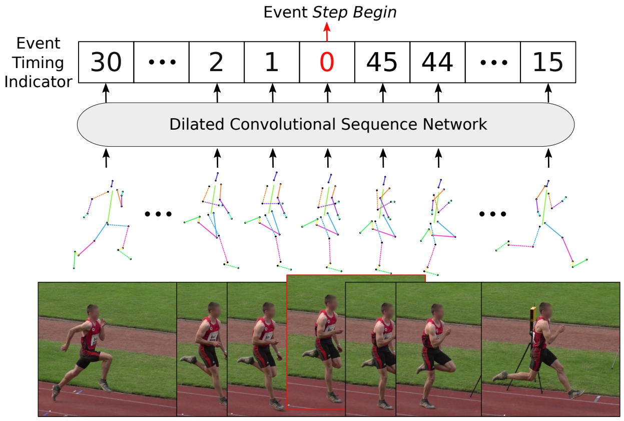

Figure 1: Our method extracts 2D pose sequences from long and triple jump recordings and predicts a timing indicator for the estimated duration until a specific event occurs in the athlete's motion. We use a dilated convolutional network to learn the sequence mapping from poses to the continuous timing indicator and extract all event occurrences on frame level.

1 INTRODUCTION

Recording, monitoring and analyzing the performance of athletes in sports is an increasingly important topic that is driven by the availability of video recordings and sensory hardware as well as the opportunities for automating the analysis with machine learning methods. Of special interest are video based methods, where athletes are tracked by one or multiple cameras to infer performance parameters and statistics in individual and team sports. In contrast to specific sensor instrumentation of athletes [9] or their surroundings [17], video based methods enable external monitoring that does not affect the athletes' performance or limit measurements to very specific training sites.

In this work we consider monocular athletics recordings, specifically in the domain of long and triple jump. Both disciplines have a distinct motion pattern with the run-up, two in-between jumps in the case of triple jump, the final jump, the flight phase and the landing in the sand pit. A single pannable camera next to the track is used to record the athlete during the whole run. Coaches use the recordings for detailed analysis by extracting key performance parameters. This process includes at least the manual selection of

MMSports '19, October 25, 2019, Nice, France

ACM ISBN 978-1-4503-6911-4/19/10…$15.00
https://doi.org/10.1145/3347318.3355525

relevant timestamps or events during the athlete's motion. It gives direct access to timing parameters, for example the step frequency. Additionally, the selected events can be paired with sparse annotations of 2D keypoints of the athlete in the respective video frames. This enables the extraction of approximate kinematic parameters, including the change in center of body mass in the steps right before the jump, angular measurements between body parts and vectorial velocities after the final jump. The information can be used to work out possible improvements in the athlete's technique and find suitable training strategies. However, the whole process of manual annotation is time consuming and limits its availability.

In this paper, we aim to automate this process with the focus on reliable detection of temporal events in athlete motion on a per-frame precision level. Based on the huge improvements in the field of automatic 2D human pose estimation with convolutional neural networks (CNNs) throughout the last years [3, 4, 14], we use 2D poses as an abstraction from the direct video content. Pose sequences are a compact description of the essence of human motion and are naturally suited to solve the event detection task at hand. Our solution to precise event detection in long and triple jump recordings is a two-step approach: First, we generate 2D pose sequences from the videos with state-of-the-art methods. Second, we process the pose sequences along their temporal axis to infer occurrences of specific events.

For the first step, we apply a variant of Mask R-CNN [14] for joint athlete detection and 2D pose estimation. We modify it to generate higher resolution keypoint estimates and adapt it to the athletics domain with a suitable set of keypoints. For the second step, we model the discrimination of discrete events in a pose sequence as a sequence translation task, where continuous event timing indicators are predicted. The timing indicators encode the temporal distance to the previous and next event in a pose sequence. This continuous encoding of discrete events is used as the learning objective for a temporal sequence CNN, which employs dilated convolutions over time to efficiently process pose sequences of variable length. The final discrete event occurrences can easily be extracted from the estimated timing indicators.

Our contributions can be summarized as follows:

- We present a system for precise temporal event detection in the motion of an athlete during long and triple jump recordings. The considered setting is rather unconstrained, with monocular recordings during training and competitions from an uncalibrated, pannable camera with varying viewpoints.
- We leverage the state-of-the-art in 2D human pose estimation and use pose sequences as an abstraction from the actual video content. We describe a simple architectural modification that leads to higher precision in keypoint detection for a customized body model.
- We propose to describe discrete event detection as a continuous timing prediction. Following the success of convolutional sequence networks on various temporal modes and tasks [6, 10, 19], we present a simple and efficient fully-convolutional architecture that can accurately predict event timings from pose sequences.

2 RELATED WORK

The work presented in this paper relates to existing literature mainly in two ways: the task to solve and the method used to solve it. We briefly review the literature on both aspects.

Event detection on human motion in sports Our notion of event detection in this paper is the task of discrete event detection on frame-level, that is, identifying the single time indices of event occurrences in a video of human motion. Most work on event detection in human motion can be more accurately described as action segmentation, where occurrence and duration of a certain action have to be identified. And while both tasks are closely related, the duration aspect in the latter task is usually more important than a frame-precise localization [32].

In the context of action segmentation, [7] use motion capture data of people who perform general physical exercises. The authors identify keyposes in the motion capture data that are characteristic for certain exercises. The occurrences of keyposes segment the overall motion into different actions. Similarly, [8] apply conditional random fields on 3D pose data from a RGB-D sensor to identify keyposes in martial arts exercises. [20] propose a method for action classification in monocular high diving recordings. They use video motion segmentation and fit a simple body model on the segmented athlete. A Hidden Markov Model is used for inferring the most probable action given the sequence of body configurations. Specifically in the athletics domain, [5] use estimated athlete velocities from motion segmentation in videos for a high-level semantic classification of long jump actions. More recently, [21] use pose similarity and temporal structure in 2D pose sequences to segment athlete motion in long jump recordings into sequential phases.

In general, there is less work to be found specifically on discrete action detection in human motion. [34] use noisy 2D pose estimates of multiple athletes in broadcasts of 100 meter races. They register lane markers from a specific camera viewpoint and align them with the 2D poses to infer time and location of ground contact for each athlete. [31] use highly specific body part detectors for swimmers in hand-held camera recordings. They extract video frames with certain body part configurations that mark the start of a swimming stroke. [36] consider the same task in specific swimming channel recordings and derive keypose occurrences from 2D pose configurations. Most similar to our work is the approach in [13, 32]. They model discrete events in periodic swimming motion as a continuous sinusoidal signal that represents the likelihood of event occurrence. They train a 3D-CNN directly on very short image sequences from monocular swimming recordings to predict the continuous event signal. In contrast, our approach is based on event timing prediction with 2D pose sequences as an intermediate motion representation, allowing for an efficient model with much longer input sequences.

Sequence models on 2D poses with CNNs Sequential 2D poses have been used in partially and fully convolutional networks for a wide variety of learning tasks. For 2D pose estimation itself, [12, 24] refine 2D pose estimates from individual video frames in recurrent CNNs with temporal supervision. Recently, 2D pose sequences are used in two-step approaches to estimate 3D poses of humans using temporal convolutions or LSTMs [18, 27, 28]. [19] use convolutional encoder-decoder networks on pose sequences –

although only in 3D – to predict human motion forward in time. [35] consider 2D pose sequences for force estimation. Lastly, [25] use pose sequences in a multi-task CNN for human action recognition in videos.

Our work We follow the success of CNN sequence models on pose information and use 2D pose sequences for subsequent discrete event detection. Compared to existing work [31, 34, 36], we avoid handcrafted features on human pose and body configuration to locate keyposes or event occurrences. Instead, we formulate event detection as a continuous sequence translation task that can be efficiently learned by a convolutional sequence network. To the best of our knowledge, such networks have not been considered for event detection in pose sequences before.

3 METHOD

In this work we focus on video-based discrete event detection in the domain of long and triple jump, but our method is applicable to general athletics and other pose events in time. The videos are taken during training as well as actual competitions and are recorded with a single pannable camera located next to the running track. Each video shows all or the majority of the run up, the jump and flight phase as well as the landing in the sand pit. The precise location of the camera along the track and its elevation can vary, leading to differences in scale and perspective of the athletes across videos. The camera is operated manually to track a single athlete. Hence, the perspective relative to an athlete also changes within a video. All video footage is recorded with the same constant frame rate. Our goal is to detect movement-related events on frame-level, based on temporal 2D pose information alone.

For this paper we consider the detection of stride-, jump- and landing-related events. Specifically, we define the events *step begin* and *step end*, which describe the beginning and end of ground contact of a foot. We do not explicitly distinguish between ground contact of the left and right foot. Our approach is mainly designed around the detection of such frequently reoccurring events during an athlete's motion. For ablation purposes we additionally define the non-repeating event *landing* as the moment of first contact of the feet with the sand pit.

While our set of detected events is exemplary, it enables the extraction of a variety of parameters: Since the beginning and end of each ground contact segments the overall motion into individual strides, stride duration and frequency can be derived directly. Additionally, approximate kinematic parameters from the 2D pose information can be inferred at those specific event timestamps.

3.1 2D pose sequences

We start by extracting 2D poses of an athlete frame-by-frame in the long and triple jump recordings. We use a modified variant of Mask R-CNN [14] to detect the athlete of interest and his/her body keypoints. The model is fine-tuned on sampled video frames with annotations following a body model with $J = 20$ keypoints. Compared to other large-scale pose datasets, this body model specifically includes the feet of the athlete, with keypoints describing the heel and the toe tips. A visualization of the body model is shown in Figure 1. The extension of the body model is crucial to detect the stride- and landing- related events.

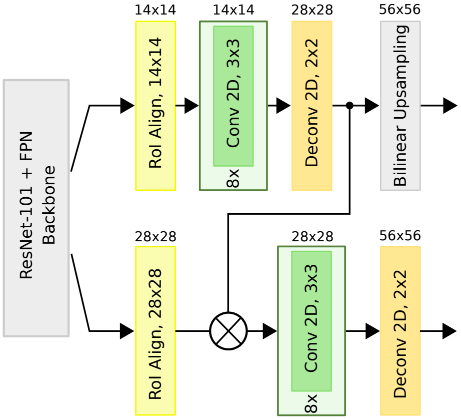

Figure 2: Modified variant of Mask R-CNN [14] with two keypoint branches. The spatial output resolution is shown above each node. Standard Mask R-CNN only uses the upper keypoint branch, where the output detection maps have an effective spatial resolution of 28×28 (followed by $2\times$ bilinear up-sampling). Our variant adds a second keypoint branch (bottom) that concatenates higher resolution backbone features and the initial detection maps, and generates refined keypoint detections at a true spatial resolution of 56×56.

Our initial experiments showed that standard Mask R-CNN leads to sub-optimal precision in the joint detections, especially for the arms and feet. The main reason is the low spatial output resolution of the Mask R-CNN architecture: First, the person of interest is detected by a Region Proposal Network (RPN) [29] on ResNet-101 [15] features with Feature Pyramid Network (FPN) resolution aggregation [23]. Then, the features in the detected spatial region are sub-sampled into a rectangular spatial grid of size 14×14. The resulting stack of feature maps is processed at the same spatial resolution by a separate network branch for keypoint prediction. Only at the very end, the features are up-sampled by a factor of two with a learned deconvolution layer, followed by regular bilinear up-sampling. The effective output resolution of the keypoint branch after the deconvolution layer is therefore only 28×28. This is roughly half the output resolution of other comparable pose estimation architectures [4, 33]. In order to resolve this lack in spatial precision, we follow the idea of repeated re-estimation and refinement of an initial set of keypoint detections [3, 26, 33], possibly on different spatial resolutions [4]: We add a second keypoint branch to the Mask R-CNN architecture that operates on double the resolution. The architectural change is depicted in Figure 2. We first combine 28×28 FPN features with the predicted output maps after the deconv layer in the default low-resolution keypoint branch. The combination of FPN features and initial keypoint predictions is then refined by a second keypoint branch at a resolution of 28×28 with an otherwise identical architecture. The output maps after the final deconvolution layer have a true spatial resolution of 56×56 without bilinear up-sampling. Both keypoint heads are trained simultaneously on the same ground truth. Our results show that this leads to an improvement in 2D pose estimation and subsequently in discrete event detection.

3.2 Event timing prediction in pose sequences

Given a video of length N, we obtain a sequence of poses $\mathbf{p} = (p_1, \ldots, p_N)$ with each $p_t \in \mathbb{R}^{J\times 2}$ describing the image coordinates of the $J = 20$ detected joints. All videos are fully annotated with event occurrences. For each event type $c \in C$, we denote the set of annotated events as $\mathbf{e}_c = \{e_{c,1}, \ldots, e_{c,E}\}$, where each $e_{c,i} \in [1, N]$ is the frame index of an event occurrence of type c. With $C = \{\textit{step begin}, \textit{step end}, \textit{landing}\}$, all $\mathbf{e}_c$ are disjoint. Therefore, we can describe the event detection task as a classification task that assigns each p_t to either one of the event types $c \in C$ or to *no event*.

3.2.1 Sequence modeling of sparse events. Modeling the task as a classification problem leads to a major class imbalance, since event annotations throughout a video are rather sparse. We propose to model event detection, specifically for reoccurring events, as a sequence translation task instead. Given a pose sequence $\mathbf{p} = (p_1, \ldots, p_N)$, we predict an event timing indicator $f_c(t)$ for every event type c and every time index $t \in [1, N]$ with

$$f_c(t) = \min_{\substack{e_{c,i} \in \mathbf{e}_c \\ e_{c,i} \geq t}} \frac{e_{c,i} - t}{t_{\max}}. \tag{1}$$

$f_c(t)$ essentially represents the normalized *forward* duration from time index t to the closest future event of type c. Normalization constant $t_{\max}$ ensures $f_c(t) \in [0, 1]$. If there is no suitable event in the future, we set $f_c(t) = 1$. In the same manner we define an event timing indicator $b_c(t) \in [-1, 0]$ that represents the *backward* duration to the closest past event of type c:

$$b_c(t) = \max_{\substack{e_{c,i} \in \mathbf{e}_c \\ e_{c,i} \leq t}} \frac{e_{c,i} - t}{t_{\max}}. \tag{2}$$

For all event occurrences $e_{c,i}$ we have

$$f_c(e_{c,i}) = b_c(e_{c,i}) = 0. \tag{3}$$

An example for the timing indicators is shown in Figure 4.

Encoding the event prediction task as the prediction of a continuous timing indicator has inherent advantages:

(1) We avoid the imbalance between event- and non-event time indices. The prediction targets are uniform in $[0, 1]$ or $[-1, 0]$.
(2) The simultaneous prediction of a forward and backward timing indicator actively encourages forward and backward propagation of information along the temporal axis of the input sequence.

3.2.2 Convolutional sequence architecture. In order to solve the event timing prediction task, we adopt a fully convolutional sequence network. Such networks have been successfully applied to various sequence-to-sequence tasks, including machine translation [10], language modeling [6] and motion prediction [19]. Our architecture takes a sequence of detected 2D poses as input, where each pose p_t is flattened into a one-dimensional vector of size $2J$. Note that the poses are predictions themselves, and thus contain imprecise detections as well as outliers. The network performs repeated convolution along the temporal axis and predicts the forward and backward timing indicators for every input pose. The timing indicators for all event types are predicted simultaneously, such that the output at each time index is a vector of size $2 \cdot |C|$. Compared to recurrent architectures (RNNs) [2], the convolutional approach has the advantage of parallelism over the temporal computations and enables a fine-grained control of the possible temporal dependencies that can be learned and leveraged by the network [1, 27]. The latter is essentially determined by the temporal receptive field of the architecture design, and can be increased with a deeper network or larger convolution kernels. We additionally employ dilated convolutions to further increase the temporal receptive field while maintaining computational efficiency and a manageable network depth and parameter number.

Our proposed network is inspired by the generic temporal convolutional network architecture (TCN) [1]. It consists of K sequential TCN blocks, starting with block $k = 1$. Each block encapsulates two dilated convolutions with kernel size w and dilation factor d. In contrast to [1] we do not use causal convolutions. This would render the prediction of $f_c(t)$ impossible. Each convolution is followed by batch normalization, rectified linear activation and dropout. The block is surrounded by a ResNet-like residual connection [15]. We increase the dilation factor with each subsequent block by a factor of two with $d = 2^{k-1}$. The K blocks are followed by a single convolution with $w = 1$ to map to the required output dimension. Figure 3 gives an overview of our main architecture. We additionally explore a deeper architecture where the second convolution in each TCN block is a non-temporal convolution with $w = 1$, similar to [27]. This results in twice the number of TCN blocks at a fixed receptive field.

With the modeling of our prediction targets, $f_c(t)$ and $b_c(t)$ only depend on the temporally adjacent event occurrences. The network therefore does not necessarily require a temporal receptive field covering the pose sequence of a complete video to infer the adjacent events during the mostly repetitive motion. We therefore limit the input of the network during training to a fixed sequence length s. No zero-padding is applied during convolutions, which otherwise has shown to have negative impact on sequence models [27, 35]. We further do not pad the input sequences, since no events are annotated at the very beginning or end of a video. Based on the architecture and the number of TCN blocks, we set the input length s equal to the temporal receptive field. The output of the network then consist of only the target values at the central sequence index $m = \lfloor \frac{s}{2} \rfloor$. Additionally, sequences in a minibatch are sampled from different videos. This counters otherwise correlated batch statistics since subsequent poses from the same video and the respective target values are highly correlated [27].

Given an input sequence of poses $\mathbf{p} = (p_1, \ldots, p_m, \ldots, p_s)$, we train the network with the averaged Huber loss $L(\mathbf{p})$ on all predicted timing indicators $\hat{f}_c, \hat{b}_c$ at the central sequence index m:

$$L(\mathbf{p}) = \frac{1}{2|C|} \sum_{c \in C} h\left(f_c(m) - \hat{f}_c(m)\right) + h\left(b_c(m) - \hat{b}_c(m)\right), \tag{4}$$

where

$$h(x) = \begin{cases} \frac{x^2}{2} & |x| \leq 1 \\ |x| - \frac{1}{2} & \text{otherwise} \end{cases} \tag{5}$$

is the Huber (smooth-L_1) loss [11, 16]. Besides dropout regularization we use standard L_2 regularization on all convolution kernel parameters.

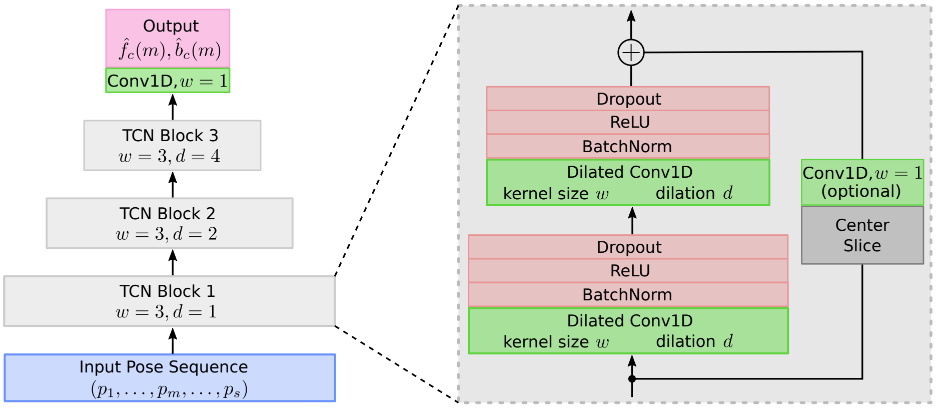

Figure 3: Base variant TCN-3 of our convolutional sequence network to predict event timing indicators. The input pose sequence is transformed by three sequential TCN blocks. For training, the sequence length s is equal to the receptive field of the network, such that the output consists of the event timing indicators for the central sequence index only. All TCN blocks have an internal size of $n = 180$ along the non-temporal axis. The final convolution after the last TCN block maps to the required output size. Since all dilated convolutions are without padding, the residual connection in each TCN block slices its input along the temporal axis before element-wise addition.

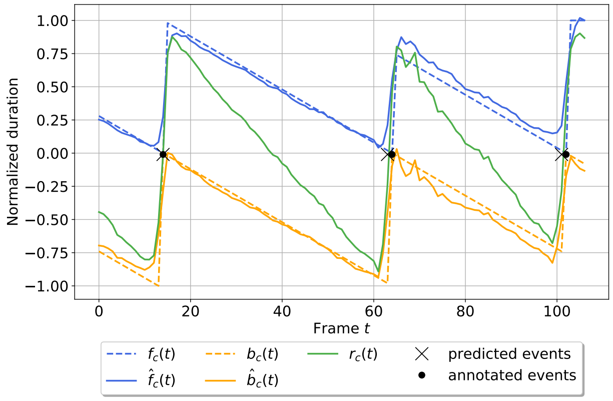

Figure 4: Example for ground truth timing indicators $f_c(t), b_c(t)$, predictions $\hat{f}_c(t), \hat{b}_c(t)$ and the derived combined indicator $r_c(t)$ on the last three steps of a long jump video (c = *step begin*). Even though the predicted timing indicators can be imperfect and noisy, the sign changes in $r_c(t)$ allow a precise event extraction.

3.2.3 Event extraction. For discrete event prediction, we first predict the continuous event timing indicators $\hat{f}_c(t), \hat{b}_c(t)$ for the complete pose sequence in a video. In contrast to training, we can process the entire pose sequence at once during inference, due to our fully-convolutional architecture. With perfect predictions, all events could be identified by $\hat{f}_c(t) = \hat{b}_c(t) = 0$. With imperfect predictions however, we have to rely on local minima/maxima close to 0 in the forward or backward timing indicators, respectively. As an alternative, both timing indicators can be combined into a single discriminative event indicator $r_c(t) = \hat{f}_c(t) + \hat{b}_c(t)$. Figure 4 depicts an example of the network output and the combined event indicator. Event occurrences can be identified by $r_c(t) \approx 0$ and a sign change from negative to positive. We extract all such time indices and perform non-maximum suppression on the l surrounding frames to avoid multiple detections of the same event.

4 EXPERIMENTAL SETUP

We evaluate our approach for event detection on video recordings of long and triple jump athletes. Our dataset consists of 167 video sequences, where 117 are used for training and validation and the remaining 50 as the test set. The recordings are taken during competitions or training and show various sports sites and athletes. The recordings have a constant frame rate of $200Hz$ and a length between 670 and 1900 frames. All videos are annotated with respect to the event types $C = \{$*step begin*, *step end*, *landing*$\}$, with an average of 19 event occurrences per video.

4.1 Details on 2D pose extraction

We use standard Mask R-CNN with a ResNet-101 backbone and FPN resolution aggregation, pretrained on the 17 joint body model in COCO [22], and extend it with our proposed high resolution keypoint head. The network is subsequently fine-tuned on a set of 3000 annotated video frames from a subset of our training videos. The annotations use a body model with $J = 20$ joints that specifically include keypoints defining the feet. We reinitialize the *deconv* layer in the keypoint heads to reflect the different keypoint set and train at a reduced batch size of 4 for a total of 140k iterations until convergence. Starting with a base learning rate of 2e-3, we reduce it by factor 1e-1 after 120k iterations. We do not train blocks C1 to C3 of the ResNet-101 backbone before the learning rate reduction. Due to the small batch and dataset size, batch normalization is not trained. The final model is applied to all video sequences in the dataset to infer 2D pose sequences. Examples of pose predictions are depicted in Figures 1 and 5.

Table 1: Results on frame-level event prediction on the long and triple jump test set videos. Except for the bottom section, all results are averaged over the reoccurring events *step begin* and *step end.* Results in the second column show the direct average timing error of the sequence network output. Other columns show the final results on discrete event prediction, with the mean frame deviation of true positives, precision, recall and F_1 score under a maximum allowed frame deviation $\Delta t \in \{1, 3\}$.

		Mean frame deviation / Precision / Recall / F_1 score							
	Avg. timing error d_c	Δt=1				Δt=3			
TCN-3, $s = 29$	0.067	**0.45**	92.3	89.9	**91.1**	**0.56**	98.7	96.2	**97.4**
TCN-4, $s = 61$	0.056	0.51	89.3	87.6	88.4	0.66	98.4	96.5	**97.4**
TCN-5, $s = 125$	**0.048**	0.55	84.1	80.8	82.4	0.8	99.0	95.1	97.0
TCN-deep-6, $s = 29$	0.069	0.48	90.3	88.5	89.4	0.61	97.9	96.0	96.9
TCN-deep-8, $s = 61$	0.055	0.52	90.3	87.8	89.0	0.65	98.6	95.9	97.2
TCN-deep-10, $s = 125$	0.049	0.56	85.8	82.6	84.2	0.76	98.7	95.0	96.8
Forward prediction $\hat{f}_c(t)$ only	–	0.49	86.8	85.6	86.2	0.66	97.2	95.7	96.4
$J = 16$ body model (no feet)	0.066	0.57	83.0	81.3	82.1	0.81	98.1	96.1	97.1
TCN 3, $s = 29$, *landing* event	0.028	0.56	68.1	64.0	**66.0**	**1.04**	95.7	90.0	**92.8**
TCN 4, $s = 61$, *landing* event	0.023	**0.52**	62.0	62.0	62.0	1.13	92.0	92.0	92.0
TCN 5, $s = 125$, *landing* event	**0.018**	0.64	52.1	50.0	51.0	1.34	91.7	88.0	89.8

4.2 Details on event timing prediction

Our TCN-based convolutional sequence network for event timing prediction is trained on the inferred 2D pose sequences from all training videos. Note that the annotated video frames for Mask R-CNN fine-tuning only cover part of the training videos. This is done on purpose to avoid overly optimistic pose predictions in the training pose sequences. The network is forced to learn on imperfect pose information. The training videos contain 2167 step events, equally distributed among *step begin* and *step end*, and a single *landing* event per video.

Unless specified otherwise, we use the following setting for training and event prediction: The network consists of $K = 3$ TCN blocks with an input sequences length $s = 29$. This leads to a total training set of ~45k different input pose sequences. The poses in each input sequence are normalized by mapping the 2D keypoint positions from image coordinates to $[-1, 1]$. The network is trained with a batch size of 100, dropout rate of 0.1 and a base learning rate of 1e-2 using the Adam optimizer for 20 epochs or until convergence on a held-out set of 17 validation videos. The learning rate is reduced by 1e-1 after 10 epochs. We use the combined event indicator $r_c(t)$ for discrete event extraction as described in Section 3.2.3, with non-maximum suppression using a window size of $l = 21$ frames. Hyperparameters, including the learning rate and batch size as well as l and t_{max} are optimized on the validation set.

4.3 Evaluation protocol

After extracting event predictions for each video, we assign every prediction to its closest ground truth event. A prediction is correct if its absolute temporal distance to the assigned ground truth does not exceed a maximum frame distance Δt. We report detection performance at maximum frame distances of $\Delta t \in \{1, 3\}$. We do not consider $\Delta t = 0$, since even annotations by humans often deviate by one frame. At the same time event detection performance usually saturates at $\Delta t = 3$.

Given a maximum frame distance, we report *precision*, *recall* and the combined F_1 score (in percentage). Additionally, we report the *mean frame deviation* of all correctly detected events (true positives). Lastly, we consider the *average timing error* d_c as the error in the event timing indicators that are directly predicted by the convolutional sequence network for a complete video:

$$d_c = \frac{1}{2N} \sum_{t=1}^{N} |\hat{f}_c(t) - f_c(t)| + |\hat{b}_c(t) - b_c(t)|. \tag{6}$$

We report d_c averaged over all test set videos. Even though this measure is not a direct indicator of event detection performance, it allows to compare the capabilities of different network variants to predict events forward and backward in time. Note that the majority of our evaluation focuses on the repetitive *step* events. Section 5.4 covers the special *landing* event.

5 RESULTS

5.1 Architectures for event timing prediction

Table 1 (top) shows test set results for different architecture variants of the convolutional sequence network. The reported scores are computed jointly on event predictions of *step begin* and *step end.* We evaluate network variants *TCN-K* with $K \in \{3, 4, 5\}$ TCN blocks and a pose input sequence length of $s \in \{29, 61, 125\}$, respectively. The TCN-3 variant shows the best results regarding F_1 score and mean frame deviation. For the strict evaluation with $\Delta t = 1$ it achieves a F_1 score of 91.1 at precision/recall of 92.3/89.9. With $\Delta t = 3$ nearly all events are correctly detected with an outstanding precision/recall of 98.7/96.2 and a mean frame deviation of 0.56 (2.8*ms*). The qualitative results in Figure 5 show that our model can handle varying viewpoints and challenging conditions during actual competitions with crowded scenes and partial occlusion by other foreground objects. The TCN-4 and TCN-5 variants with increasingly longer input sequences and wider dilation lead to a decrease in performance: For strict $\Delta t = 1$ evaluation, precision and recall drop notably and lead to a loss of up to −8.7 in F_1 score. For $\Delta t = 3$, performance is relatively stable across the architectures, but with an increase in the mean frame deviation of up to +0.24 (+1.2 ms). Conversely, we see that the average timing error d_c of the directly predicted timing indicators decreases with longer input sequences, with the optimum at $s = 125$. This is mainly due to the fact that the network variants with a larger temporal receptive field can better predict the timing of more distant events. However for event extraction, we only require precise timings around each event occurrence $e_{c,i}$ with $\hat{f}_c(e_{c,i}) \approx \hat{b}_c(e_{c,i}) \approx 0$. The network variants with less dilation seem to retain a higher temporal precision, leading to better discrete event predictions. It shows that shorter pose sequences are already discriminative enough to decide on the motion phase of an athlete. A more global view on the overall motion with longer input sequences leads to a loss in precise temporal event localization.

For our deeper network variant with double the number of TCN blocks (*TCN-deep-K*), the same effect can be observed: declining performance at $\Delta t = 1$ for larger input sequences and stable results at $\Delta t = 3$. Overall, the deeper network variants with more parameters and additional computational overhead have no benefit over the regular TCN-K models.

5.2 Combined forward and backward timing prediction

A key element of our proposed sequence modeling is the simultaneous prediction of forward and backward event timing indicators. It forces the network to actively propagate information forward and backward along the temporal axis of the input pose sequence. To validate this design, Table 1 (mid-upper) shows ablation results where the TCN-3 variant is trained to predict only the forward timing indicator $f_c(t)$. Note that this also limits the extraction of discrete events to the predicted indicator $\hat{f}_c(t)$: Since event occurrences are ideally identified by $\hat{f}_c(t) = 0$, we extract local minima close to 0 and remove redundant event predictions with non-maximum suppression. The results show a major drop in $\Delta t = 1$ performance with −4.9 in F_1 score compared to the forward + backward TCN-3 model. And even for $\Delta t = 3$, where previous results across different architectures were rather stable, we observe a drop of −1.0 in F_1 score. This shows the effectiveness of simultaneous forward and backward timing prediction. In combination, both objectives form a better learning target for the sequence network and enable more precise event detections.

5.3 Influence of body model and 2D pose fidelity

With our two-step approach to discrete event detection using intermediate 2D pose sequences, any missing or wrong motion information in the pose sequences has a direct influence on event detection. Our choice of body model explicitly includes keypoints defining the feet of the athlete, to ensure that stride-related events can be precisely located. Table 1 (mid-lower) shows the results in event detection when reducing the pose estimates to $J = 16$ keypoints, where the heel and toe tip keypoints are excluded. For high temporal precision with $\Delta t = 1$ we observe a large drop in event detection performance, with a reduction of −9.0 in F_1 score. This confirms the huge advantage of 2D poses with a specialized body model that describes the body parts directly related to the relevant motion events. If less temporal precision is required ($\Delta t = 3$), the results show that a simpler body model (similar to the ones found in large-scale pose datasets) suffices to obtain comparable results.

Table 2: Effects of 2D pose extraction with standard Mask R-CNN and our variant with high-resolution keypoint predictions on pose estimation and event detection performance.

	Pose Estimation		Event Detection
	PCK@0.1	PCK@0.2	F_1 Score ($\Delta t = 1$)
Mask R-CNN [14]	79.9	91.7	88.6
+ high-res. keypoints	**83.9**	**92.6**	**91.1**

Besides the body model, the motion information in pose sequences depends on the quality and precision of the keypoint predictions. Our variant of Mask R-CNN is specifically designed to improve spatial precision and overall pose estimation performance. Table 2 shows pose estimation results on a set of 900 annotated frames from the test set videos. We report pose estimation performance with the *percentage of correct keypoints* PCK@α, where a keypoint detection is considered correct if it is located within a fraction α of the torso diameter from the ground truth location [30]. Our architectural modifications improve the performance of Mask R-CNN by +4.0 for high precision keypoint detections with PCK@0.1. This shows the effectiveness of keypoint re-estimation on a higher spatial resolution within Mask R-CNN. For lower keypoint precision with PCK@0.2, a smaller gain of +0.9 can be observed. Finally, Table 2 also shows event detection performance when our timing prediction network is trained on 2D poses from both Mask R-CNN variants. The results confirm that an increase in spatial pose estimation precision directly translates to an improvement in temporal precision for event detection: Pose sequences from our improved Mask R-CNN architecture lead to an increase of +2.5 in F_1 score for the *step begin* and *step end* events, when evaluated on the strict temporal precision of $\Delta t = 1$. Note that this improvement only comes from an architectural modification in Mask R-CNN, but otherwise no change in training data.

5.4 Non-repeating events

Even though the design of our overall approach is focused on the repeating motion during the run-up and jump phases, we also evaluate how well the landing in the sand pit can be detected. This event only occurs once in every video. Training examples for this event are very limited in our dataset, making it difficult to learn a generalizing event detection model. At the same time, most of the poses p_t in a video are temporally distant from the single landing event, leading to $f_{landing}(t) = 1$ and $b_{landing}(t) = -1$ for the majority of time indices. This results in a clear imbalance in prediction targets. Table 1 (bottom) shows results on *landing* event detection. As expected, the strict evaluation at $\Delta t = 1$ shows a large drop in performance compared to the step-related events, with the best F_1 score of only 66.0. In its current setting, our model is not able to reliably detect the landing on a frame-precise level. Interestingly,

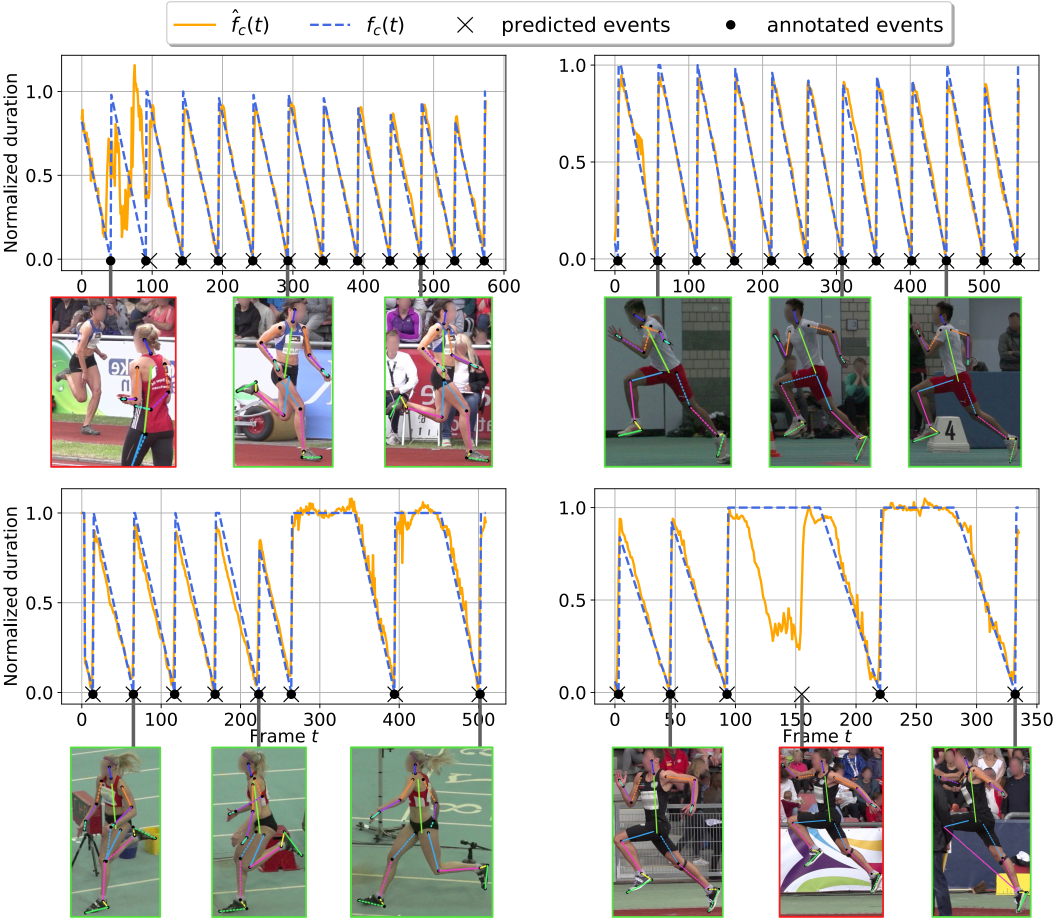

Figure 5: Qualitative examples of event detection for *step begin* (left) and *step end* (right) on test set videos. The top and bottom rows show examples for long jump and triple jump, respectively. To avoid clutter, we only show ground truth and prediction for the forward event timing indicator $f_c(t)$. Our model achieves stable and precise event detections throughout most of the videos. Errors mostly occur under keypoint missdetections over a long period of time (top left).

performance quickly recovers at the relaxed maximum frame deviation of $\Delta t = 3$: With a precision/recall of 95.7/90.0, the best model achieves a F_1 score of 92.8. This result is more comparable to the performance on the *step* events, although the mean frame deviation is roughly doubled. Across different architectures, longer input pose sequences again lead to decreasing temporal precision. The results show that in general our proposed approach can also be applied to detect non-repeating events in athlete motion. But the very few training examples and the imbalance in prediction targets proof to be a limiting factor for precise temporal localization.

6 CONCLUSION AND FUTURE WORK

In this paper, we have presented a two-step approach to automatic frame-precise discrete event detection in long and triple jump. We leverage the state-of-the art in 2D human pose estimation to obtain 2D pose sequences as an intermediate representation of an athlete's motion. Our proposed representation of discrete, reoccurring events with event timing indicators proof to be a suitable learning objective for sequence translation with dilated convolutional neural networks. Our best architecture can efficiently transform imperfect 2D pose sequences into continuous event indicators that allow highly precise discrete event extraction throughout most of the recordings and under challenging conditions. It can easily be adapted to different sets of keypoints and motion events.

Despite the very convincing results on the reoccurring stride-related events, our approach seems to be less suitable for non-repeating events where training examples are rather scarce. We aim to improve our learning objective representation to better fit such unique and non-repeating events. Additionally, we plan to incorporate the recent advances in weakly-supervised 3D human pose estimation for better viewpoint independence via 3D pose alignment.

ACKNOWLEDGMENTS

This work was funded by the Federal Institute for Sports Science based on a resolution of the German Bundestag. We would like to thank the Olympic Training Center Hessen for collecting and providing the video data.

REFERENCES

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *CoRR* abs/1803.01271 (2018). arXiv:1803.01271 http://arxiv.org/abs/1803.01271

[2] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[4] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018. Cascaded Pyramid Network for Multi-Person Pose Estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[5] Chuan Wu, Yu-Fei Ma, Hong-Jiang Zhan, and Yu-Zhuo Zhong. 2002. Events recognition by semantic inference for sports video. In *Proceedings. IEEE International Conference on Multimedia and Expo*, Vol. 1. 805–808 vol.1.

[6] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, 933–941. http://dl.acm.org/citation.cfm?id=3305381.3305478

[7] Pablo Fernández de Dios, Qinggang Meng, and Paul WH Chung. 2013. A machine learning method for identification of key body poses in cyclic physical exercises. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 1605–1610.

[8] Claudio Marcio de Souza Vicente, Erickson R Nascimento, Luiz Eduardo C Emery, Cristiano Arruda G Flor, Thales Vieira, and Leonardo B Oliveira. 2016. High performance moves recognition and sequence segmentation based on key poses filtering. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1–8.

[9] Benedikt Fasel, Jörg Spörri, Julien Chardonnens, Josef Kröll, Erich Müller, and Kamiar Aminian. 2018. Joint Inertial Sensor Orientation Drift Reduction for Highly Dynamic Movements. *IEEE Journal of Biomedical and Health Informatics* 22, 1 (Jan 2018), 77–86. https://doi.org/10.1109/JBHI.2017.2659758

[10] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, 1243–1252. http://dl.acm.org/citation.cfm?id=3305381.3305510

[11] Ross Girshick. 2015. Fast R-CNN. In *The IEEE International Conference on Computer Vision (ICCV).*

[12] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. 2016. Chained Predictions Using Convolutional Neural Networks. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 728–743.

[13] Kohei Hakozaki, Naoki Kato, Masamoto Tanabiki, Junko Furuyama, Yuji Sato, and Yoshimitu Aoki. 2018. Swimmer's Stroke Estimation Using CNN and MultiLSTM. *Journal of Signal Processing* 22, 4 (2018), 219–222.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask R-CNN. In *The IEEE International Conference on Computer Vision (ICCV).*

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[16] Peter J Huber et al. 1973. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics* 1, 5 (1973), 799–821.

[17] Shariman Ismadi Ismail, Hiroyuki Nunome, Fatin Farhana Marzuki, and Izzat Su'aidi. 2018. The Influence of Additional Surface on Force Platform's Ground Reaction Force Data During Walking and Running. *American Journal of Sports Science* 6, 3 (2018), 78–82.

[18] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. 2018. Propagating LSTM: 3D Pose Estimation based on Joint Interdependency. In *The European Conference on Computer Vision (ECCV).*

[19] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. 2018. Convolutional Sequence to Sequence Model for Human Dynamics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[20] Haojie Li, Jinhui Tang, Si Wu, Yongdong Zhang, and Shouxun Lin. 2010. Automatic Detection and Analysis of Player Action in Moving Background Sports Video Sequences. *IEEE Transactions on Circuits and Systems for Video Technology* 20, 3 (March 2010), 351–364. https://doi.org/10.1109/TCSVT.2009.2035833

[21] Rainer Lienhart, Moritz Einfalt, and Dan Zecha. 2018. Mining Automatically Estimated Poses from Video Recordings of Top Athletes. *International Journal of Computer Science in Sport* 17, 2 (2018), 94 – 112. https://content.sciendo.com/view/journals/ijcss/17/2/article-p94.xml

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). arXiv:1405.0312 http://arxiv.org/abs/1405.0312

[23] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[24] Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. 2018. LSTM Pose Machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[25] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2018. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*. Springer, 483–499.

[27] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[28] Mir Rayat Imtiaz Hossain and James J. Little. 2018. Exploiting temporal information for 3D human pose estimation. In *The European Conference on Computer Vision (ECCV).*

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 91–99.

[30] Ben Sapp and Ben Taskar. 2013. MODEC: Multimodal Decomposable Models for Human Pose Estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[31] Long Sha, Patrick Lucey, Sridha Sridharan, Stuart Morgan, and Dave Pease. 2014. Understanding and analyzing a large collection of archived swimming videos. In *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 674–681.

[32] Brandon Victor, Zhen He, Stuart Morgan, and Dino Miniutti. 2017. Continuous Video to Simple Signals for Swimming Stroke Detection With Convolutional Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.*

[33] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4724–4732.

[34] Kentaro Yagi, Kunihiro Hasegawa, Yuta Sugiura, and Hideo Saito. 2018. Estimation of Runners' Number of Steps, Stride Length and Speed Transition from Video of a 100-Meter Race. In *Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports (MMSports'18)*. ACM, New York, NY, USA, 87–95. https://doi.org/10.1145/3265845.3265850

[35] Dan Zecha, Christian Eggert, Moritz Einfalt, Stephan Brehm, and Rainer Lienhart. 2018. A Convolutional Sequence to Sequence Model for Multimodal Dynamics Prediction in Ski Jumps. In *Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports (MMSports'18)*. ACM, New York, NY, USA, 11–19. https://doi.org/10.1145/3265845.3265855

[36] Dan Zecha, Christian Eggert, and Rainer Lienhart. 2017. Pose Estimation for Deriving Kinematic Parameters of Competitive Swimmers. *Electronic Imaging* 2017, 16 (2017), 21–29. https://doi.org/doi:10.2352/ISSN.2470-1173.2017.16.CVAS-345