All Keypoints You Need: Detecting Arbitrary Keypoints on the Body of Triple, High, and Long Jump Athletes

Katja Ludwig   Julian Lorenz   Robin Schön   Rainer Lienhart

Chair for Machine Learning and Computer Vision, University of Augsburg

{katja.ludwig, julian.lorenz, robin.schoen, rainer.lienhart}@uni-a.de

Abstract

Performance analyses based on videos are commonly used by coaches of athletes in various sports disciplines. In individual sports, these analyses mainly comprise the body posture. This paper focuses on the disciplines of triple, high, and long jump, which require fine-grained locations of the athlete's body. Typical human pose estimation datasets provide only a very limited set of keypoints, which is not sufficient in this case. Therefore, we propose a method to detect arbitrary keypoints on the whole body of the athlete by leveraging the limited set of annotated keypoints and auto-generated segmentation masks of body parts. Evaluations show that our model is capable of detecting keypoints on the head, torso, hands, feet, arms, and legs, including also bent elbows and knees. We analyze and compare different techniques to encode desired keypoints as the model's input and their embedding for the Transformer backbone.

1. Introduction

Analyzing athletes during trainings or competitions is popular among all sports disciplines. In team sports, typical applications are tactics analyses or the tracking and evaluation of athlete's running paths and the ball during a game. Coaches and athletes of individual sports use video analyses to precisely track the body position. They use these methods to assess the performance of the athletes, to derive improvements, and develop appropriate training routines. The key to analyze the body position is mostly to identify the locations of specific keypoints on the body of an athlete. With these keypoints, exact movements and derived parameters can be evaluated. In this paper, we focus on the disciplines of triple, high, and long jump. Athletes of these disciplines are highly interested in their step frequency, speed, and body posture during the in-run and the jump phase.

Typically, only professional athletes have access to such evaluations as they require a large effort. 2D Human Pose

Figure 1. Two detection results of arbitrary keypoints on images of our *jump-broadcast* dataset using our model, visualized with three equally spaced lines to both sides of each body part including the outer boundary in pure color and the central line in white with a color gradient from the boundary to the central line.

Estimation (HPE) models can help to improve the availability of such video analyses, because they can automate the keypoint detection part which is commonly the most time consuming task. However, the fixed set of keypoints that standard 2D HPE models provide is not sufficient for some kinds of analyses. Adding more keypoints to a fixed set requires annotating all new keypoints in most images, which is again very time consuming and the resulting model is still limited to a (larger) fixed set of keypoints. In order to remove such restrictions, recent works [10, 11] have proposed a technique to detect arbitrary keypoints on the limbs of humans. They adapt the Vision Transformer (ViT) [2] based method TokenPose [7] by replacing the learned tokens for each fixed keypoint with a query embedding. This query embedding encodes the keypoint that should be detected. Query embeddings are generated from human readable query inputs. After various Transformer layers, the output of the model corresponding to these query embeddings is converted to small heatmaps from which the final keypoint detection is obtained. However, these methods are currently limited to the limbs, which we try to overcome in this paper. Figure 1 shows that our model is capable of detecting arbitrary points on the head, torso, arms, and legs in-

cluding hands and feet. It also correctly estimates keypoints on bent limbs like elbows and knees. The contributions of this work can be summarized as follows:

- We enlarge the area of the body for which arbitrary keypoints can be detected by the feet, hands, torso, and head. We further improve this model such that it can detect keypoints on bent elbows and knees correctly.

- We propose different techniques for the model to encode the desired target keypoints. We encode the head either based on a reference line or based on an angle. Additionally, we use an encoding based on coordinates of a normalized pose. We further apply different embedding methods to the encoded input.

- We release the *jump-broadcast* dataset, a new dataset with 2403 annotated triple, high, and long jump athletes sampled from 27 hours of 26 different TV broadcast videos. All images are annotated with 20 keypoints. The dataset further contains 1797 automatically generated segmentations masks. The dataset is available here: https://www.uni-augsburg.de/en/fakultaet/fai/informatik/prof/mmc/research/datensatze/

- Experiments on the *jump-broadcast* dataset and a second triple and long jump dataset prove that our methods are capable of detecting any desired keypoint on the body of athletes. We improve the evaluation scheme of previous work by using more keypoints and provide a detailed evaluation of the performance of keypoints located on the different body parts. Our best approach for the jump-broadcast dataset is available here: https://github.com/kaulquappe23/all-keypoints-jump-broadcast

2. Related Work

2D HPE is a popular technique for computer vision aided analyses in individual sports. Recently, Transformer [14] based 2D HPE architectures are gaining popularity, while CNN architectures like the High Resolution Net (HRNet) [17] are still very common. TokenPose [7] is a Transformer architecture that achieves the best results with the first three stages of an HRNet as feature extractor, but there exists also a pure Transformer variant. All variants make use of the ViT [2] architecture, which cuts an image or a feature map into small patches which serve as the input sequence to the Transformer. ViTPose [19] proves that pure ViT based HPE models are also capable of achieving SOTA scores by adding a decoder with deconvolutions after the ViT layers. The HRFormer [20] architecture combines the ideas of the HRNet and the ViT and uses ViT layers while maintaining branches of different feature resolutions like the HRNet.

Leveraging the idea of focusing on the part of the image where the person is located, Zeng et al. [21] cluster tokens of less important image areas like the background, while keeping many tokens for important areas. Ma et al. [12] follow a similar idea by deleting the tokens of unimportant image areas. They achieve a more lightweight architecture as a consequence. Apart from that, focusing on multi-person pose estimation, Shi et al. [13] propose a fully end-to-end framework based on Transformers.

Video analyses during training or competitions are common for most professional sports athletes. Analyses in team sports often involve tracking and identification of the ball (and alike) and the players on the field in order to analyze their trajectories, e.g. in (ice) hockey [4, 15], soccer [1], volleyball, football, or basketball [4]. The tracking task is challenging in the domain of sports, as the athletes are similarly dressed, move fast and are often occluded or out of view of the camera. In individual sports, typical analyses involve estimating keypoints on the human body and sports equipment. Liu et al. [9] detect the badminton shuttle and reconstruct its 3D trajectory from monocular videos. Table tennis stroke types and the poses of the athletes are further detected by Kulkarni et al. [6]. Stepec et al. [16] use estimated poses of ski jumpers and their trajectories in order to automatically generate the style score. Hudovernik et al. [5] estimate the poses of competing Jiu-Jitsu athletes to automatically detect combat positions and derive the scores of the athletes. Since the footwork is really important in fencing, Zhu et al. [23] estimate the poses of fencers and classify fine-grained footwork actions of the athletes.

3. Jump-Broadcast Dataset

We release the *jump-broadcast* dataset to enable a public benchmark on arbitrary keypoint detection for triple, high, and long jump athletes. We have collected 26 videos of competitions from broadcast TV footage, summing up to 27 hours of video material. 9 videos cover triple jump competitions, 8 videos long jump competitions and the remaining 9 videos high jump competitions. A total of 193 different male and female athletes are present in the video footage. The sports sites, lighting conditions and image quality vary throughout our dataset. Moreover, it contains a lot of extreme poses, especially during the jump phase. We select the frames by sampling approx. 5 equidistant frames from each jump (including in-run and jump phase) and each camera perspective. Slow motion replays are mostly recorded with a different camera and therefore seen as a new camera perspective. We select 2403 images in total and annotate them with the following 20 keypoints: head, neck, left/right shoulder, left/right elbow, left/right wrist, left/right hand, left/right hip, left/right knee, left/right ankle, left/right heel, left/right toe tip. Furthermore, the annotations include information whether a frame corresponds to a slow motion

Figure 2. Cropped example images from the *jump-broadcast* dataset. Segmentation masks are displayed as an overlay. Annotated keypoints are visualized in white. These examples show the variety of poses in our dataset, including occlusions, front and side views, the in-run, and extreme poses during the jump phase.

replay and the name of the athlete as is presented in the TV broadcast. We split the dataset in 1805 images for training, 576 images for testing and 122 images for validation in such a way that each athlete is only included in a single subset, even if they have participated in multiple competitions.

We use the DensePose [3] framework from detectron2 [18] to automatically generate segmentation masks for our dataset. Since some images are very blurry and the athletes perform extreme poses in comparison to everyday activities, some masks are completely or partly wrong. We sort out the worst segmentation masks by hand but keep masks that are partly correct. The advantage of our approach is that it can deal with partly correct segmentation masks in most cases. In the end, we keep 1797 segmentation masks, 1338 belonging to the training set, 97 to the validation set and 362 to the test set containing the head, torso, left/right upper arm, left/right forearm, left/right hand, left/right thigh, left/right lower leg and left/right foot. Figure 2 visualizes some images with generated segmentation masks.

4. Method

All variants are based on the TokenPose [7] architecture. In a first step, this architecture takes an image and feeds it through the first three stages of an HRNet for feature extraction. These feature maps are then split into visual feature patches and embedded to create visual tokens via a linear projection. The visual tokens are fed jointly with keypoint query tokens through multiple Transformer layers. In the end, the output of the ViT corresponding to the keypoint query tokens is transformed to heatmaps via a shared MLP. The final keypoint coordinates are then retrieved from the heatmaps with the DARK [22] method. In contrast to the pure TokenPose architecture, we do not learn representations for the keypoint query tokens, but use transformations to embed human readable encodings in the token domain. Our method can be applied to any TokenPose variant.

4.1. Ground Truth Generation

For the limbs, we follow the strategy described in [10,11] to generate ground truth keypoints. At first, we draw a vector from one enclosing keypoint to the other, which we will call v_e. Second, an orthogonal line l_o to v_e is created and the furthest points c_l and c_r that lie on l_o and the segmentation mask are retrieved. c_l is always located on the left side relative to the orientation of v_e and c_r on the right side. Arbitrary keypoints are then located on the line between the so-called intersection points c_l and c_r. We further add the constraint that c_l and c_r need to be part of a coherent area of the segmentation mask in order to deal with small errors in our automatically generated segmentation masks, like the left thigh in the fourth image of Figure 2. We incorporate the same technique for the feet, using the toe tip and the heel as the keypoints that enclose the body part. For the torso, we use the neck as the first enclosing keypoint and the virtual keypoint in the middle of the hip keypoints as the second one. Regarding the hands, this technique has the drawback that it only detects keypoints that lie between the enclosing keypoints. If we would use the hand and wrist keypoint as enclosing keypoints, our model would not be able to detect the finger tips. Since the finger tips are not annotated, we extend the line through the wrist and the hand keypoint beyond the hand keypoint to the boundary of the hand segmentation mask and use this point instead of the hand keypoint as the second enclosing keypoint during ground truth generation. See Section 4.1.1 for more details.

4.1.1 Head

Standard keypoints that are usable as enclosing keypoints for the head are neck and the head keypoint itself. In contrast to other datasets, where the head keypoint is located at the top of the head, the head keypoint is located in the middle of the head in our datasets. Therefore, we cannot use it as an enclosing keypoint, in this case we would only be able to detect keypoints on the lower half of the head. Hence, we use the strategy already described for the hands, which is also visualized in Figure 3 on the left. At first, the line v_e through the neck and head keypoint is created and extended to the head top k_t and the chest k_b, visualized with orange dots in Figure 3. We select a random intermediate point k_i on v_e (visualized with a green dot in Figure 3), draw the orthogonal line l_o to this line, determine the
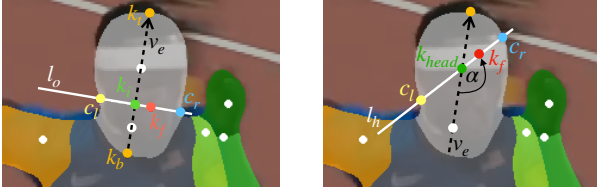
Figure 3. Two versions of generating ground truth keypoints on the head. On the left, the line through the head and neck keypoint is extended to the head boundary (orange), an orthogonal line to this line is drawn (white) and a keypoint on this line between the boundary points of the segmentation mask (blue and yellow) is chosen (red). On the right, a line (white) rotated around the head keypoint (green) at a random angle is generated and a keypoint on this line between the boundary points of the segmentation mask (blue and yellow) is selected (red).

intersection points c_l and c_r with the boundary of the segmentation mask and select a random point k_f between k_i and $c_{l/r}$. We call this the *extension* method.

Since the head is rather round, the creation process close to the head top seems counterintuitive. The distance between c_l and c_r is decreasing and even approaching 0, the closer k_i is to k_t. Hence, we propose a second strategy to generate keypoints on the head. We use the head keypoint as the center, choose a random angle $\alpha \in [0, 2\pi)$ and rotate a line l_h counterclockwise around the center according to α, while $\alpha = 0$ corresponds to v_e. Then, we continue as before with detecting c_l and c_r as the intersection of l_h with the boundary of the segmentation mask and randomly choosing a point k_f between the head keypoint and $c_{l/r}$, which is visualized in Figure 3 on the right. c_r is used for angles $\alpha \in [0, \pi)$ and c_l for $\alpha \in [\pi, 2\pi)$. Since we rotate for a certain angle, we refer to this as the *angle* method.

4.1.2 Bent Limbs: Elbows and Knees

With the keypoint generation strategy on the limbs as used in [10, 11], arbitrary keypoints on the upper arm, forearm, thigh, and lower leg could be detected if they lie in between

Figure 4. Visualization of the anchor generation at a bent knee. We generate a line with half the bending angle β through the knee keypoint, visualized in yellow. Then, we determine the intersection of that line with the segmentation mask boundary (orange and green) and select the anchor point within the acute angle, so the anchor is the orange point in this image.
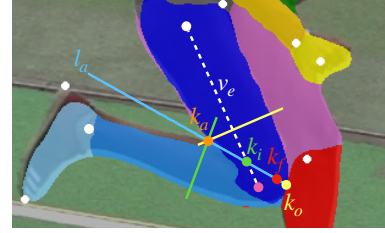

Figure 5. Visualization of the keypoint generation on a bent knee. The anchor point is visualized in orange, the line l_a in blue. The intersection point k_i is colored green and the point k_o yellow. The final keypoint is visualized in red. The line l_a always lies somewhere between the yellow and the green line.

the enclosing keypoints. This works well in most cases, but not if the limbs are strongly bent, which happens frequently in the case of triple, high, and long jump. Therefore, we adapt the generation strategy to fit also the case of bent elbows and knees. For that purpose, we determine the point on the inner side of the bent joint at first, which we call the *anchor point* or k_a in the following. To retrieve that point, we calculate the bending angle β of the joint, which is the angle enclosed by the line through the hip and the knee and the line through the knee and the ankle or the angle enclosed by the line through the shoulder and the elbow and the line through the elbow and the wrist. Then, we generate a line l_i through the knee or elbow keypoint with half of the bending angle. We determine the intersections of that line with the boundary of the segmentation mask and set the anchor k_a as the point on the side with the acute angle. We unify the segmentation mask for the lower and upper body part in this case. See Figure 4 for details. In the next step, visualized in Figure 5, we rotate a line l_a for a random angle around the anchor point, starting in a $90°$ angle to the upper body part (yellow line in Figure 5) and stopping at a $90°$ angle to the lower body part (green line in Figure 5). We determine the intersection point k_i of l_a with the vector v_e through the enclosing keypoints of the corresponding body part (upper or lower one) and chose a random final keypoint k_f either between k_i and k_a or between k_i and k_o. We do not discriminate between the segmentation mask for upper and lower body part in the whole process since the boundary is often detected rough and it is somehow not clearly defined where the thigh/upper arm ends and the lower leg/forearm begins. With this strategy, arbitrary points on bent limbs can be generated.

4.2. Query Encoding

With the described strategies, it is possible to generate arbitrary ground truth points. However, it is impossible to maintain an infinite set of keypoints in the way that a heatmap is generated for every possible keypoint, like described in [10, 11]. Therefore, we use a different strategy

and tell the model which keypoints it should detect with a special query encoding which is part of the model's input. Since ViT architectures can deal with inputs of various length, it is possible to query for an arbitrary number of keypoints. In this paper, we evaluate two fundamentally different query encodings and two nuances for one variant.

4.2.1 Vectors

The fist approach uses multiple vectors to encode the desired keypoint. Let the dataset contain n keypoints, then the first vector called *keypoint vector* has length n and the second vector called *thickness vector* has length 3. For the limbs and the feet, we follow the strategy described in [10] to fill the vectors. Let k_1, k_2 be two keypoints enclosing a body part and k_i be the orthogonal projection of the desired keypoint on the line l_o between k_1 and k_2. Then $k_i = p \cdot k_1 + (1 - p) \cdot k_2$. We set the entries belonging to k_1 and k_2 in the keypoint vector to p and $1 - p$, respectively. For the torso, we set the keypoint vector entry belonging to the neck to p and the entries belonging to the left and right hip joint to $0.5(1 - p)$, since we use the middle of the two hip keypoints as the reference point. In case of the hands and the extension method for the head, where we extend the line through the reference keypoints k_1 and k_2 to the points k_t and k_b, $k_i = p \cdot k_t + (1 - p) \cdot k_b$, but we set the entries in the keypoint vector in the same way as before. The thickness vector is created similarly, let the final keypoint be $k_f = q \cdot c_{1/2} + (1 - q) \cdot k_i$, then we set the middle entry of the thickness vector to $1 - q$ and the first or last entry to q, depending if c_l or c_r is chosen [10].

If the angle method is used for the head, this approach is not applicable as the creation logic is completely different. Therefore, we introduce a third vector called the *angle vector*. It has just one entry which is set to 0 in case of all other body parts, meaning that it is not used. For the head, its value indicates the rotation angle α in percent, starting with the line through head and neck keypoint and rotating counterclockwise around the head keypoint like described in Section 4.1.1. This means that the lines l_h for rotation angle percentages $0 < p <= 0.5$ and $0.5 + p$ are identical. They differ in their corresponding intersection point. c_r is used for percentage p and c_l for $0.5 + p$. The final keypoint can now be calculated as $k_f = q \cdot c_{l/r} + (1 - q) \cdot k_{head}$, and we set the thickness vector to $[0, 1 - q, q]$ no matter which intersection point is used as this information is already encoded in the angle vector. We further set the entry of the keypoint vector corresponding to the head keypoint to 1.

For elbows and knees, we adapt the encoding slightly. In these cases, k_i is not projected perpendicular (along l_o) on the line between k_1 and k_2, but along the line l_a and we further use l_a to determine the intersection points c_l and c_r and the thickness percentage q.
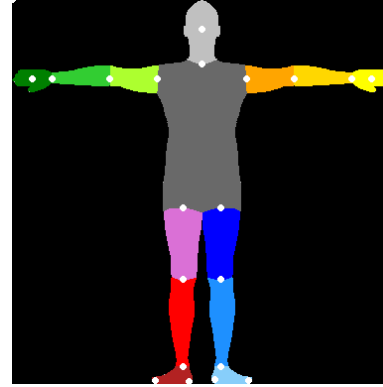

Figure 6. Visualization of the normalized pose. All used body parts are colored and the fixed keypoints are visualized in white.

4.2.2 Normalized Pose

The second approach involves a normalized human pose, similar to [10]. The normalized human pose contains the keypoints and the body part segmentation masks of a human in a T-shaped pose, visualized in Figure 6. We use feet turned outwards and hands downwards such that the maximum area of these body parts is visible in the segmentation masks. Each desired keypoint is now represented with the normalized x- and y-coordinate of this pose. Normalized coordinates mean that the upper left corner has coordinates $(0, 0)$ and the lower right corner $(1, 1)$. Hence, each keypoint is characterized by only two values in this encoding approach.

4.3. Keypoint Token Embedding

Each keypoint encoding, no matter which of the three described options is used, needs to be converted to a token of the same size as the visual tokens in order to be compatible with the ViT. This process is called *embedding*. The straightforward way is to use a linear layer to convert a vector of arbitrary length to a vector with the desired length. In case of the vector approach, we have more than one vector in the encoding. Hence, we have the option to concatenate the vectors to one vector at first and then embed them, or to embed them at first to tokens with a half or a third of the embedding size and concatenate them afterwards. Furthermore, apart from using a single linear layer, we try to enhance the embeddings by adding more layers and ReLU operations.

5. Experiments

The base architecture for all our experiments is TokenPose-Base [7] with an input image size of 192×256, an embedding dimension of 192 and a conversion of the final embeddings to heatmaps of size 48×64. Athletes are cropped before the images are fed through the network. We

pretrain our model on the COCO dataset [8] and finetune it on our individual sports related datasets. Random flipping, random rotation of up to $45°$, random scaling in the range of $[0.65, 1.35]$, and color jitter are applied as augmentation techniques during training.

5.1. Evaluation Metrics

We use the Percentage of Correct Keypoints (PCK) and the Percentage of Correct Thickness (PCT) as evaluation metrics. Both metrics calculate the percentage of all desired keypoints that the network has correctly predicted at a threshold t. The PCK uses the distance between the left shoulder and right hip keypoints times the threshold as the maximum distance for a correct prediction. We use $t = 0.1$ and $t = 0.05$ in our evaluations, which corresponds to approx. 6cm and 3cm, respectively. The PCT calculates the differences between the predicted thickness and the ground truth thickness for keypoints on all body parts analogous to the keypoint generation by subtraction or addition depending if the ground truth and predicted keypoint are located on the same or different sides of the intermediate point/the head keypoint. Correct predictions are the ones with thickness differences below t. We use $t = 0.2$ in our evaluations.

[10] uses a random number of up to 50 freely chosen keypoints per image during evaluation. [11] increases their evaluation strategy to a fixed number of 200 randomly created keypoints. In this paper, we aim to evaluate the full range of possible keypoints together with the different encoding and embedding strategies for certain body parts. Hence, on each body part, we create a specific set of evaluation keypoints: we use thickness $0, 0.5$ and 1 and create 25 equally spaced keypoints with these thicknesses to both sides of the body part, resulting in 125 keypoints per body part. During evaluation, we treat elbows and knees as individual body parts, too. Hence, if all body parts are completely visible, this results in a total number of 2250 keypoints per image. The benefit of this strategy is that it is reproducible and hence provides a comparable metric for future work. It also weights all body parts equally and allows a separate comparison of the results for each body part.

5.2. Collapsing Elbow and Knee Keypoints

Elbows and knees connect the upper arm with the forearm and thigh with the lower leg, including them in our method enables retrieving continuous keypoints along the whole arms and legs. Especially in these cases, it is very important to interpret v_e as a vector and define c_l and c_r relative to its direction. This ensures that c_l and c_r are always located on the same side of the upper body part (thigh or upper arm) and the lower body part (lower leg or forearm). If c_l and c_r swap the side between the lower and upper body part, we can encounter a collapse of the keypoints around the swapping location, visualized in Figure 7.
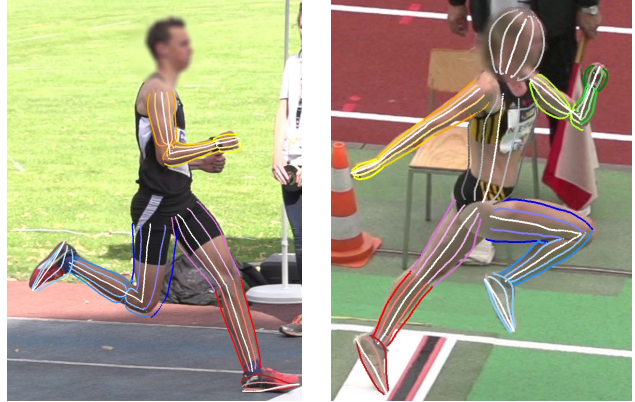

Figure 7. Collapsing points around the right elbow and knee (first image) and the left elbow (second image) in the case that c_l and c_r swap sides between the lower and upper body part.

5.3. Triple and Long Jump Dataset

We first evaluate our methods on the triple and long jump dataset. The quality of its images is higher than that of the jump-broadcast dataset since the cameras were installed specifically for the analysis tasks and not for TV broadcasts. It contains 4101 training images, 1462 testing images and 464 validation images from various sports sites, indoor and outdoor locations, and from a variety of different athletes during triple and long jump competitions or trainings. It is labeled with the same 20 keypoints as the jump-broadcast dataset. Segmentation masks are also obtained with detectron2 [18] and kept for all images.

We execute four experiments with different embedding strategies using the vector encoding with the extension technique for the head, five experiments with the angle technique and three experiments with the normalized pose encoding. The results are displayed in Table 1. Overall, the different strategies achieve similar results. All are capable of detecting arbitrary keypoints on all body parts. Generally, the approaches with the vector encoding achieve slightly better results in all body parts. Even the extension strategy and angle strategy achieve similar scores for the head keypoint. However, the angle strategy improves the scores for the feet, while the performances for all body parts on the upper body (torso and arms) slightly drop. The number of layers in the embedding process does also not make a large difference independent from the encoding type, which is in contrast to the results reported in [10]. For the extension strategy and the normalized pose approach, the single layer embeddings achieve even the best overall scores. Regarding the number of layers before concatenation during the embedding, we encounter that later concatenations slightly improve the results for both encodings. The detection scores vary largely among the different body parts. The hands seem the most challenging body parts, since the

Enc.	Emb.	Con.	PCK@0.1	PCK@0.05	PCT	head	torso	u.arm	elbow	f.arm	hand	thigh	knee	l.leg	foot
V-E	1	1	94.6	82.5	69.7	77.0	65.7	67.6	71.4	64.0	58.6	75.3	75.7	73.4	70.1
V-E	2	1	94.5	82.2	69.6	76.9	68.6	66.7	70.6	64.9	60.7	74.0	74.4	72.4	69.7
V-E	2	0	94.7	82.5	69.3	76.2	68.6	66.2	70.0	63.8	59.3	75.2	73.7	72.9	69.8
V-E	3	1	94.4	82.2	69.4	75.1	66.4	66.5	70.3	64.5	60.5	74.4	74.9	72.8	69.7
V-A	1	1	94.6	82.4	70.8	77.5	68.8	66.3	70.9	64.3	59.9	75.9	75.2	73.7	77.6
V-A	2	1	94.7	82.5	70.3	77.7	68.4	65.4	70.3	64.4	61.0	74.9	74.5	73.0	75.5
V-A	2	0	95.0	82.8	71.0	77.6	68.8	66.6	71.5	65.4	60.8	75.5	75.3	73.4	77.1
V-A	2	2	94.9	82.3	70.1	76.6	67.9	65.6	69.5	65.0	60.3	74.9	74.0	73.1	75.7
V-A	3	1	94.6	82.1	69.7	75.6	66.7	65.9	70.0	63.4	59.4	74.8	73.4	72.4	76.4
NP	1	-	94.1	80.4	65.7	71.4	62.0	57.8	66.0	58.1	53.7	71.9	72.9	71.2	71.2
NP	2	-	93.8	78.4	63.5	67.9	59.5	53.8	65.3	56.5	52.9	69.6	70.7	69.9	67.9
NP	4	-	93.8	79.4	65.2	67.3	59.7	55.4	68.0	58.9	53.3	71.1	73.2	71.6	69.8

Table 1. Recall values for the triple and long jump test set in % at PCK@0.1 and PCK@0.05 and PCT@0.2. These scores are evaluated on the test set with the fixed keypoints and on the described 2250 keypoints per body part as far as these keypoints exist in the image. The first column indicates the used encoding: *V* indicates the vector approach, either with the head extension strategy (symbolized with *V-E*) or the head angle strategy (*V-A*). *NP* refers to the normalized pose approach. The second column names the number of layers used for the embedding and the third column the number of layers that is executed before concatenation of the vectors. The third table section contains the average metric results over all keypoints in the test set, the fourth section lists the PCT scores separately for each keypoint type, left and right keypoints are combined.

PCT is at most 60.7%. Thighs, knees and head are the body parts with the best scores over all experiments. These results can be explained by the fact that these body parts generally lead to larger (and therefore also thicker) segmentation masks, which make it easier for the network to learn and detect precise keypoints. However, it is surprising that the model achieves a better performance for the elbow keypoints than for the upper arm and the forearm, although the thickness does not change here. Furthermore, it is noticeable that the extension strategy performs worse on the feet than on the legs, while the angle strategy performs better and the normalized pose approach performs similar over all experiments. The best overall PCT and PCK@0.05 score is achieved with the vector encoding and the extension approach and a single linear layer embedding. Qualitative results for this dataset are visualized in the first two images in Figure 8.

5.4. Jump-Broadcast

For the jump-broadcast dataset, we execute two experiments with the vector encoding and the extension strategy, three experiments with the angle strategy and two experiments with the normalized pose encoding (chosen according to the best experiments from the triple and long jump dataset results). The results are shown in Table 2. Since the quality of the images is worse than for the triple and long jump dataset, the scores are generally lower. The effect that the results of the angle strategy improve the score for the feet can be observed for this dataset as well. The improvement is even larger, with an absolute increase of
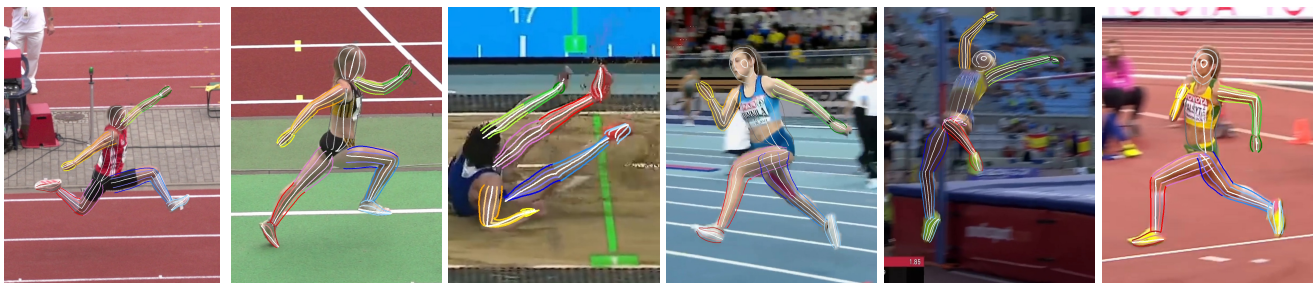

Figure 8. Detection results for images of the triple and long jump dataset (first two images) and the *jump-broadcast* dataset (last four images), visualized with equally spaced lines to both sides of each body part including the outer boundary in pure color and the central line in white with a color gradient from the boundary to the central line. The first two images are generated with the vector encoding using the extension strategy and the last four images with the angle strategy for the head in order to show the differences between the strategies. Occluded/overlapping body parts are omitted for clarity.

Enc.	Emb.	Con.	PCK@0.1	PCK@0.05	PCT	head	torso	u.arm	elbow	f.arm	hand	thigh	knee	l.leg	foot
V-E	1	1	90.4	71.1	61.2	61.9	48.3	63.6	57.3	51.9	44.9	67.1	67.3	74.3	61.9
V-E	2	1	90.8	71.6	63.6	66.2	54.0	68.0	59.1	55.7	47.4	69.3	67.5	75.4	64.6
V-A	1	1	90.8	71.3	63.3	63.5	50.7	67.2	58.7	54.6	45.4	67.3	66.9	75.5	70.5
V-A	2	1	90.9	71.1	63.7	63.8	48.6	67.1	58.3	56.2	47.1	68.5	67.7	74.3	71.3
V-A	2	2	91.0	71.6	63.8	65.2	52.2	67.1	57.5	55.6	47.0	68.8	67.7	73.7	72.2
NP	1	-	89.0	68.0	57.3	47.4	39.1	59.0	52.4	46.4	38.3	62.9	64.4	71.3	67.4
NP	4	-	89.2	68.2	58.2	52.4	42.3	59.1	53.1	46.9	39.5	64.2	64.3	71.8	68.3

Table 2. Recall values for the triple and long jump test set in % at PCK@0.1 and PCK@0.05 and PCT@0.2. These scores are evaluated on the test set with the fixed keypoints and on the described 2250 keypoints per body part as far as these keypoints exist in the image. The first column indicates the used encoding: *V* indicates the vector approach, either with the head extension strategy (symbolized with *V-E*) or the head angle strategy (*V-A*). *NP* refers to the normalized pose approach. The second column names the number of layers used for the embedding and the third column the number of layers that is executed before concatenation of the vectors. The third table section contains the average metric results over all keypoints in the test set, the fourth section lists the PCT scores separately for each keypoint type, left and right keypoints are combined.

over 7% for the best variants. The score for the other keypoints are similar or slighly lower regarding the best variant of the extension encoding and the angle encoding. Using more layers shows a marginal improvement for this dataset, whereby there is no significant difference observable for the experiments with different concatenations. Keypoints on the hand body part stay the most challenging ones in this dataset. However, the scores of the elbow and the forearm are significantly lower than the score of the upper arm for this dataset. The best scores are achieved for the lower leg, the foot, the thigh, the upper arm and the head. The normalized pose approach achieves the lowest overall scores for this dataset as well. The largest drop is observable for keypoints on the head and torso body part with an absolute difference of approy. 10%. The scores for the upper arm, forearm, and hand further drop more than 5%. Additionally, for this dataset in contrast to the triple and long jump dataset, the angle strategy achieves the best overall results (PCT, PCK@0.05 and PCK@0.1) with a two layer embedding and the concatenation after these two layers. Example detections for this method are displayed in the last four images in Figure 8.

6. Conclusion

In this paper, we propose a method to detect arbitrary keypoints on the whole human body for athletes of triple, long, and high jump. We introduce a new dataset consisting of images from these disciplines in order to provide a publicly available dataset for future comparisons. This *jump-broadcast* dataset comprises 2403 images from 27 hours of 26 different TV broadcast videos showing 193 athletes. We obtain 1797 segmentation masks of the head, torso, l./r. upper arm, l./r. forearm, l./r. hand, l./r. thigh, l./r. lower leg, and l./r. foot.

We extend the methods introduced in [10, 11] to make our model capable of estimating arbitrary keypoints on all body parts. To generate ground truth keypoints for the hands, which have no second annotated enclosing keypoint, we extend the line through wrist and hand keypoint to the boundary of the segmentation mask to obtain a second point. For the head, we use either the same technique, or we rotate a line around the head keypoint. We further calculate the points on the inner side of the elbow and knee joints and rotate around these points in order to generate arbitrary points on the elbow and knee, since these joints are often heavily bent during jumps and could not be estimated correctly until now. We evaluate our model with different encodings of the arbitrary keypoints, either as vectors with the extension strategy for the head or the angle strategy, or as keypoint coordinates of a normalized human pose. In the vector case, we introduce a third vector in case of the angle strategy and evaluate the performance depending on the number of embedding layers used before we concatenate the embeddings of the single vectors. Evaluations on the triple and long jump dataset and our newly introduced jump-broadcast dataset show that the results for all variants are generally similar. The normalized pose approach achieves slightly lower scores regarding all experiments. Concatenating the vector embeddings later results in minimal better scores. For both datasets, keypoints on the hand are the most challenging. The extension and angle approach for the head have strengths for different keypoints. For the triple and long jump dataset, the extension approach with a single layer embedding achieves the best overall score while the angle approach with a two-layer embedding leads to the best performance for the jump-broadcast dataset. Hence, the method proposed in this paper is capable of detecting arbitrary keypoints on the whole human body of triple, long, and high jump athletes, including bent elbows and knees.

References

[1] Anthony Cioppa, Silvio Giancola, Adrien Deliege, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Marc Van Droogenbroeck. Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3491–3502, 2022. 2

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[3] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. 2018. 3

[4] Hsiang-Wei Huang, Cheng-Yen Yang, Samartha Ramkumar, Chung-I Huang, Jenq-Neng Hwang, Pyong-Kun Kim, Kyoungoh Lee, and Kwangju Kim. Observation centric and central distance recovery for athlete tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 454–460, 2023. 2

[5] Valter Hudovernik and Danijel Skocaj. Video-based detection of combat positions and automatic scoring in jiu-jitsu. In *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*, MMSports '22, page 55–63, New York, NY, USA, 2022. Association for Computing Machinery. 2

[6] Kaustubh Milind Kulkarni and Sucheth Shenoy. Table tennis stroke recognition using two-dimensional human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4576–4584, 2021. 2

[7] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. *arXiv preprint arXiv:2104.03516*, 2021. 1, 2, 3, 5

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[9] Paul Liu and Jui-Hsien Wang. Monotrack: Shuttle trajectory reconstruction from monocular badminton video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3513–3522, 2022. 2

[10] Katja Ludwig, Daniel Kienzle, and Rainer Lienhart. Recognition of freely selected keypoints on human limbs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3531–3539, 2022. 1, 3, 4, 5, 6, 8

[11] Katja Ludwig, Daniel Kienzle, Julian Lorenz, and Rainer Lienhart. Detecting arbitrary keypoints on limbs and skis with sparse partly correct segmentation masks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 461–470, 2023. 1, 3, 4, 6, 8

[12] Haoyu Ma, Zhe Wang, Yifei Chen, Deying Kong, Liangjian Chen, Xingwei Liu, Xiangyi Yan, Hao Tang, and Xiaohui Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In *Computer Vision– ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 424–442. Springer, 2022. 2

[13] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11069–11078, 2022. 2

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[15] Kanav Vats, William McNally, Pascale Walters, David A Clausi, and John S Zelek. Ice hockey player identification via transformers and weakly supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3451–3460, 2022. 2

[16] Dejan Štepec and Danijel Skočaj. Video-based ski jump style scoring from pose trajectory. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 682–690, January 2022. 2

[17] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[18] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 3, 6

[19] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022. 2

[20] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *arXiv preprint arXiv:2110.09408*, 2021. 2

[21] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022. 2

[22] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020. 3

[23] Kevin Zhu, Alexander Wong, and John McPhee. Fencenet: Fine-grained footwork recognition in fencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3598, 2022. 2