IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 March 2023, accepted 30 March 2023, date of publication 7 April 2023, date of current version 20 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265595

APPLIED RESEARCH

DiveNet: Dive Action Localization and Physical Pose Parameter Extraction for High Performance Training

PRAMOD MURTHY[1,2], BERTRAM TAETZ[2,3], ARPIT LEKHRA[1], AND DIDIER STRICKER[1,2]

[1]Department of Augmented Vision, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), 67663 Kaiserslautern, Germany
[2]Department of Augmented Vision, German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
[3]Department of IT and Engineering, International University of Applied Sciences (IU), 99084 Erfurt, Germany

Corresponding author: Pramod Murthy (pramod.murthy@dfki.de)

This work was supported in part by the Federal Ministry of Education and Research (BMBF), Germany, through the Project DECODE, under Grant 01IW21001; and in part by the German Swimming Federation (DSV).

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT The tremendous progress of deep convolution neural networks has shown promising results on the classification of various sports activities. However, the accurate localization of a particular sports event or activity in a continuous video stream is still a challenging problem. The accurate detection of sports actions enables the comparison of different performances, objectively. In this work, we propose the DiveNet action localization module to detect the springboard diving sports action in an unconstrained environment. We used Temporal Convolution Network (TCN) over a backbone feature extractor to localize diving actions, with low latency. We estimate the divers center of mass (COM) trajectory and the peak dive height using the temporal demarcations provided by the action localization step via the projectile motion formula. In addition, we train a DiveNet pose regression network, which extends the Unipose architecture with direct physical parameter estimation, i.e COM and 2D joint keypoints. We propose a new homography computation method between the diving motion plane and the image-view for each dive. This enables the representation of physical parameters in metric scale, without any calibration. We release the first publicly available diving sports video dataset, recorded at 60 Hz with a static camera setup for different springboard heights. DiveNet action localization achieves an accuracy of 95% with a single frame latency (< 25 ms). The DiveNet pose regression model shows competitive results around 70% PCK on different diving pose datasets. We achieve COM accuracy of 6 pixels, dive peak height sensitivity of 20 cm and mean joint angle errors around 10 degrees.

INDEX TERMS Deep learning, diving sports pose, action localization, sports analytics.

I. INTRODUCTION

The body pose and motion analysis of sports athletes is an emerging research area in computer vision and deep learning [1], [2], [3], [4]. The analysis of sports videos is a challenging task because the body postures and manoeuvres involved in sports such as diving, gymnastics and balance-beam are complex. In addition, monocular based human pose estimation often suffer from self-occlusion and motion blur due to high-speed nature of actions that are performed [5], [6], [7]. These problems often cause many markerless motion captures to provide inferior results [8]. Sports video analysis aims to provide objective measures to compare the athlete's performance, e.g. during training. For a given sports performance, video analysis can help to improve the athlete's technique with real-time feedback [9], [10], [11],

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

[12], [13]. Thus, it assists sport trainers in technique analysis, thereby increasing the overall efficiency of the coaching process [14]. Existing sports analysis is largely analyzed manually to observe the movement of the diver's body parts during training. The feedback provided after manual analysis is time-consuming, laborious and error-prone [15]. In recent years, alternative technologies are proposed [16], but they still have limitations in using them to provide accurate and robust results [15]. So far, existing analysis techniques often focused on diving classification are challenges due to high variability in the scenes. The variability occurs due to various factors such as people in the background, image resolution and different viewing positions [17].

Our work primarily focuses on competitive springboard diving training. Thus, we differ from recent works on diving action classification [5], [6], [7] as we solely focus on accurate diving localization. We localize the diving motion and segment the video in time, emphasizing low latency segmentation bounds rather than performing general action classification for a motion clip. This is particularly important to compare specific motion segments of a motion during training that are defined via certain human poses, as suggested e.g. in [18]. Hence, we focus on segmentation of a video to localize a particular set of motions in time throughout a continuous video by detecting the frame representing the beginning and end of the (diving) motion [19] as shown in Figure 1. In other words, it is a precise detection of the start frame and end frame of diving sports action in an untrimmed video. On the other hand, diving action classification assigns one of 48 diving classes based on the combination of four attributes: 1) takeoff, 2) somersaults, 3) twists, and 4) dive positions while entering the water [6], [7].

Action localization is an essential step before action classification in a continuous video stream or long untrimmed videos. It eliminates the need for manual effort to localize motion in continuous live video or untrimmed sports videos. To this end, we have the following key contributions, in this work:

- We propose a novel TCN based low-latency solution for diving action localization in untrimmed videos.
- We present a 2D pose estimation model with accurate center of mass computation.
- We propose a novel homography computation method between the diving motion plane and the image-view to estimate the diving peak height and accurate motion trajectory.
- We release the DSV diving dataset with action localization demarcation and its corresponding 2D pose keypoints for the research community. The dataset is available for research at https://av.dfki.de/ murthy/divenet/.

II. RELATED WORKS

There is a surge in sports video analysis in recent years due to the new advances in deep learning methods and the availability of high computing resources. A common theme for learning complex temporal relationships between the actions in long untrimmed videos is to encode videos in an end-to-end manner using pre-trained 2D or 3D CNNs [20], [21], [22]. Sequence-to-sequence models use these learned high-level features to localize the action.

A. ACTION PROPOSAL NETWORKS

The action proposal networks typically use search strategies such as selective search sampling to produce sequence proposals [23]. The bounding boxes also known as *Tubelets* instead of super-pixels use super-voxels. Thereby eliminating the issue of linking boxes from one frame to another. Taking inspiration from R-CNN [24] and Faster R-CNN [25], Temporal Action Localization network (TAL-Net) exploits temporal context from the videos for both proposal generation and classification [26], [27]. Action proposal based networks are effective but require an exhaustive search of the complete video, making them computationally expensive and infeasible, especially in very long videos [26].

B. GRAPH NEURAL NETWORKS

The learning of contextual relationships of sports actions in videos can be modelled using graph-based methods. A *context graphs* represents the similarities among videos and the relationship between segments of a video [28]. The graph is learned using a context walk with a fixed number of steps and creating a conditional distribution over all super voxels. The Segment Action proposals are predicted using Conditional Random Fields. Thus, context graphs reduce the search space and avoid a need for a sliding window over a complete action. Soomro et al. [28] learns context relationships and represents them by a graph for each video, where super-voxels form the nodes and directed edges capture the spatial relations between them. Graph based methods work by modelling similarities and relationships between segments of a video to be able to form a graph and cluster similar information making it easier to classify them. The Graph Convolutional Networks (GCN) is also used to capture the spatial and temporal pattern in the data [29], [30]. A spatio-temporal graph convolution is applied on pose estimation from videos to generate a higher-level feature maps using a graph. These feature maps are then classified to corresponding action categories using a classifier with softmax outputs. The GCN can also be used to explicitly model appearance and motion similarities between video moments with a weakly supervised method [30]. A stacked version of GCN, Stacked Temporal GCN (ST-GCN) represents elements related to the actions such as actors, objects, etc. as nodes to better characterize the complex actions in videos [31]. The nodes are connected along the spatial and temporal dimensions instead of having body joints as nodes of the graph.

C. RECURRENT NEURAL NETWORKS MODELS

The Recurrent Neural Networks (RNN) are one of the most popular methods for modelling temporal information

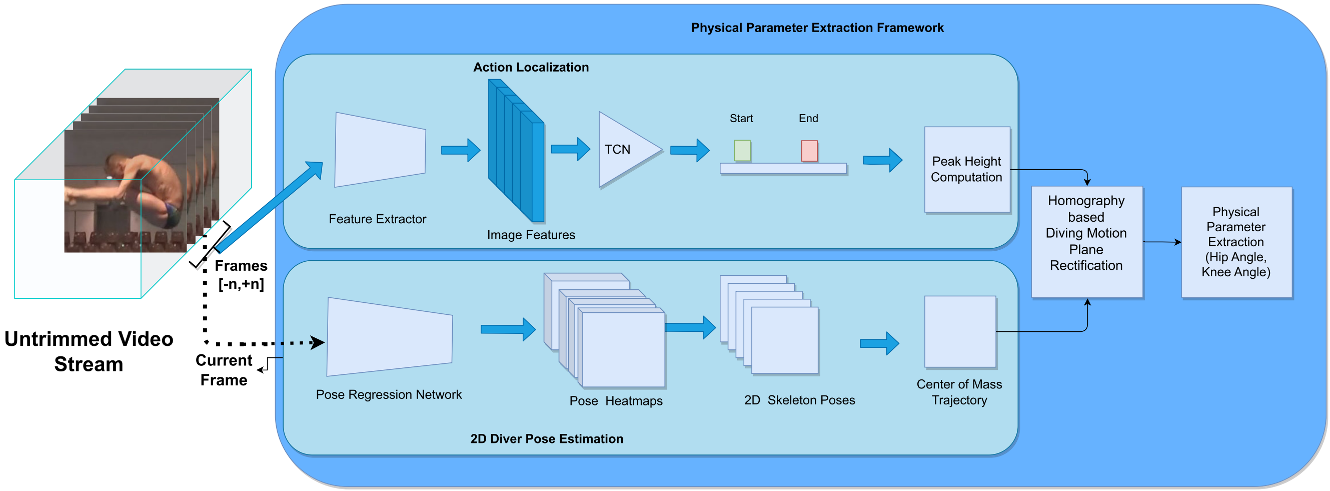


FIGURE 1. Illustration of physical parameter extraction framework from diving action video.

in sequences [32], [33], [34]. A MultiLSTM model further expands the temporal relation by using multiple input and output connections of an LSTM [35]. The work of [36] used a novel two-stream feedback network instead of a standard LSTM layer. The two-stream consists of the upper stream, which focuses on the interpretation of the frames, and the lower stream that models the temporal relations. The Kanojia et al. [6] presented a novel attention guided LSTM architecture with an encoder and decoder network for localizing actions in diving motion. The attention network utilizes the global context from the encoder and learns to focus on the diver during the dive without any such supervision. However, apart from being hard to correctly train, recurrent neural networks only capture implicit relationships between certain high motion actions [37].

D. TEMPORAL CONVOLUTION NETWORKS

While RNN variants and graph-based models are hard to train [37], they also tend to capture only implicit relationships between specific motion actions and have a limited span of attention. On the other hand, TCNs can process long-range patterns due to the kernels sharing weight for all time steps. The work by Lea et al. [38] was the first to use TCN networks for action detection and segmentation. They presented two TCN variants. First, Encoder-Decoder TCN hierarchically modelled the actions using temporal convolutions, pooling and upsampling. Second, Dilated TCN used a deep stack of dilated convolutions with skip connections. They showed that both TCN models outperformed their Bidirectional LSTM network [39] and were much faster to train. More recent variants of TCN for action detection are PDAN [40], MS-TCN [41] and Multi-tower TCN [42]. MS-TCN extend the work of Lea et al. [38] by stacking multiple dilated TCNs to form an ensemble a multi-stage networks. Each stage produces an initial prediction that is refined by the next one and by computing the loss after each stage. Multi-tower TCN also uses multiple TCN networks, but instead of forming an ensemble, the networks are stacked in parallel. The outputs from all the TCN towers (having different receptive fields) are fused before passing through the softmax layer. The intuition behind a multi-tower structure is to use multiple receptive fields to processes information at various temporal scales to deal with different event frequencies. One drawback of using convolutions is that they assign the same importance to each local feature in the kernel, preventing kernels from selecting the region of interest effectively. In an attempt to remedy this, Dai et al. [40] proposed the Pyramid Dilated Attention Network (PDAN) that employs dilated attention layers to allocate attentional weights to local frames in the kernel. Inspired by the success and better performance of TCNs and its ability to learn short-term and long-term temporal relations over other sequence modelling methods [43], we develop a temporal convolution network for diving action localization in this work.

E. SPORTS POSE

There has been a tremendous success in estimating human skeleton pose in images using deep learning methods. The works based on stacked hour glass newtorks [44], [45], [46] achieved remarkable performance on different datasets. OpenPose [47] employs Part Affinity Fields (PAF) to support bottom-up estimation. The authors of DeepHR Net [46] exploit multi-scale high-resolution networks to improve the feature representation. The UniPose model [48] based on Waterfall Atrous Spatial Pooling architecture, achieves state-of-art-results on several pose estimation metrics. The recent VitPose model [49] explores the potential of plain and non-hierarchical vision transformers [50] and provides a simple yet effective vision transformer baseline for pose estimation tasks. A very recent, also based on transformers, uses a pose regression network to map images to keypoint co-ordinates, without resorting to intermediate representations [51].

A lot of methods address human motion analysis in sports videos, differently. Based on self-supervised learning, the

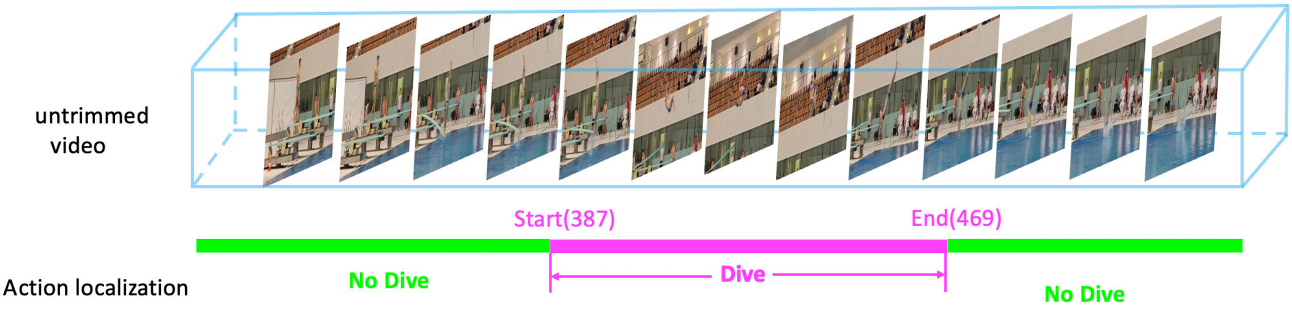


FIGURE 2. Illustration of diving action localization in an untrimmed video.

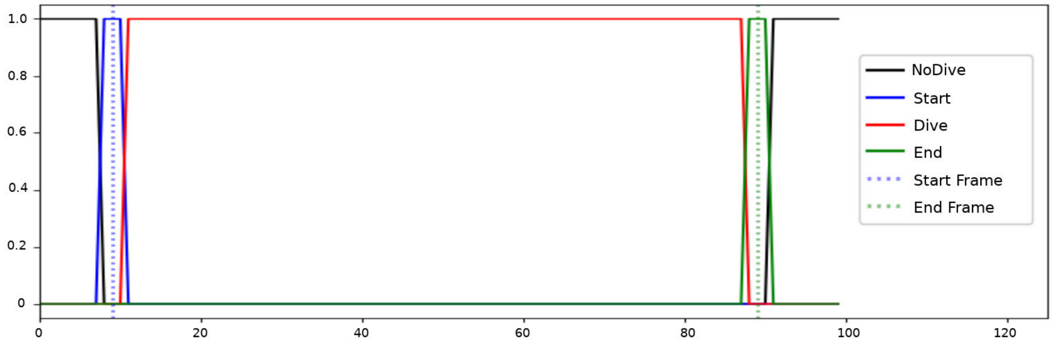


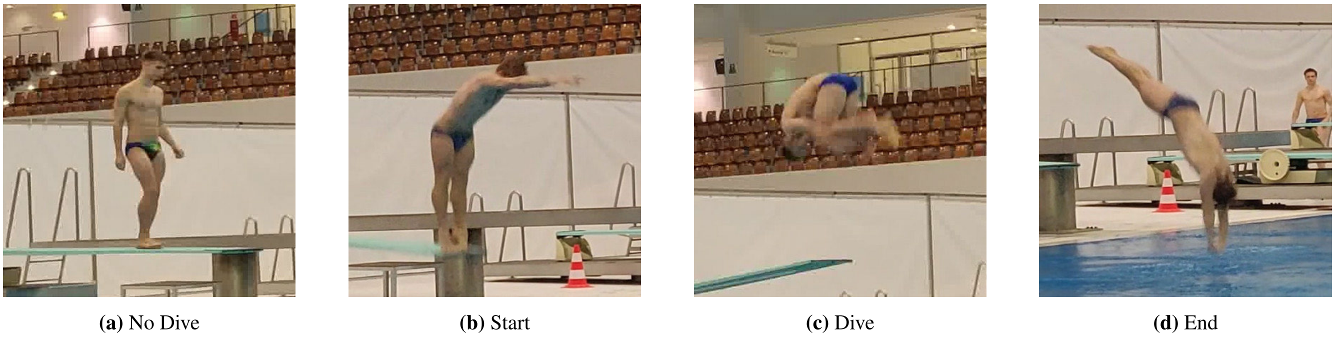
FIGURE 3. Action Localization using class labels (NoDive, Start, Dive, End) for video frames. The dotted vertical blue line and green line represent the start frame and end frame of the diving action respectively.

suggested method in [52] employs pseudo labels as a self-supervised training strategy, together with a pseudo label filtering method, in the disciplines of triple and long jump. The study, described in [53], leads to the dataset *Kanoya*, which is composed of numerous series of videos of gymnasts doing acrobatic motions and proposes several refining approaches to train OpenPose [47] for extreme human poses in sports. The method offered by [54] introduces Video Pose Distillation (VPD), a weakly supervised technique for learning features for novel video domains, such as individual sports that challenge pose estimation and shows the benefits of VPD on four varied sports video datasets: diving, floor exercises, tennis and figure skating, with fine grained action labels. The recent works of [55] combines the use of computer vision algorithms and fully convolutional neural networks. The proposed marker-less 2D swimmer pose estimation approach estimates the pose of a swimmer during exercise while guaranteeing adequate measurement accuracy. While the work of [56] presents a procedure-aware technique for action quality assessment, trained via a new temporal segmentation attention module, it also constructs a new fine-grained dataset, termed FineDiving, created on various diving events with detailed annotations on action procedures. The work described in [57], provides an alternative unsupervised representation learning technique based on videos captured by a single RGB camera with a focus on diving sports. First, the model uses a spatial transformer network to identify the person across all frames and then encodes the subject into time-variant and time-invariant components. They sample with temporally close, distant and intermediate frames, given a reference frame.

While these methods focuses on poses, our goal is to extract a subset of physical parameters from diving action videos as accurate as possible. We use our methodology to segment untrimmed videos with low latency in order to locate specific motions in time, regress 2D joints and COM and estimate parameters such as the dive peak height as well as the precise COM motion trajectory.

III. DIVING ACTION LOCALIZATION

The diving sport involves complex body movements performed by an athlete in a short time (2-5 seconds). We apply the task of temporal action localization for diving motion. First, the dive is demarcated by detecting the start frame and end frame of the motion as shown in Figure 2. The *start* frame is a representative frame where the force exerted by the athlete to the springboard is zero. The *end* frame of the motion is when the athlete touches the water. Next, we select three frame step function to represent the boundary classes for each diving motion. Additionally, we define two more classes, namely "*NoDive*" and "*Dive*". *NoDive* class comprise all the frames not representing the diving motion. These frames usually correspond to the diver standing on the board or entering the water. The *Dive* class represents all frames where the diver is in the air during the motion. The class representation for the diving motion



(a) No Dive

(b) Start

(c) Dive

(d) End

FIGURE 4. Images from the recorded DSV diving dataset. Figure (a), (b), (c) and (d) shows examplar frames of a diving video representing the classes NoDive, Start, Dive, End, respectively.

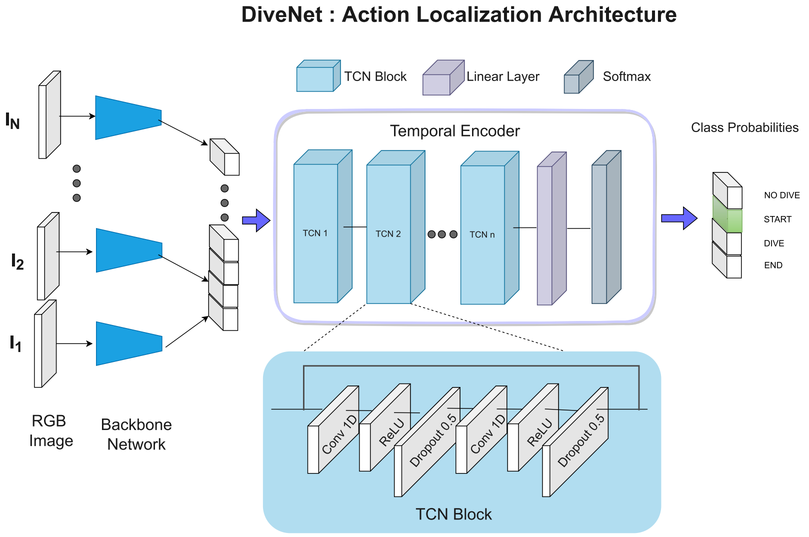


FIGURE 5. Diving action localization - TCN block architecture.

classification is as shown in Figure 3. Examples of diver pose representing different classes in diving motion are shown in Figure 4.

Figure 5 illustrates the overall architecture of the diving action localization framework. We first locate the person with a square bounding box. We crop the person represented in the image and provide it as an input (I) to the Feature Extractor Network. The extracted features for each image are concatenated and fed to Temporal Convolutional Network (TCN) to get the final class scores. The input image is cropped with the detected bounding boxes containing the diver. The cropped images are passed through feature extractor backbone network, which extracts high-level features from the images. Given a video sequence $V = \sum_{t=1}^{N}\{I_t\}$ of length N, the feature extractor networks extracts features F_t for each cropped frame I_t. The N features are split into temporal windows of length T and used as input for the temporal encoder. The temporal encoder consists of multiple TCN blocks. The higher the number of TCN blocks leads to higher receptive field i.e. the network has larger context and can see farther in time. The output from the last TCN block is T features for each item of the sequence, which are concatenated and passed through multiple 1D convolutional layers to down sample the feature size to IR^{512}. Finally, the softmax function is applied over the outputs of linear layers. The linear layer outputs represent the confidence score for each class, and the softmax layer squashes and normalizes the score of each class to be between 0 and 1.

We use two different types of backbone feature extraction networks for action localization:

- ResNet: The ResNet-18 is used as a backbone feature extractor to segment diving frames in the specified classes. The feature vector of 512 represents a single frame as input to the classifier.
- Human Mesh Recovery: We use HMR model from [1] as a backbone feature extractor to achieve the motion segmentation. We use feature vector size of 2048 to represent the pose of the person as input.

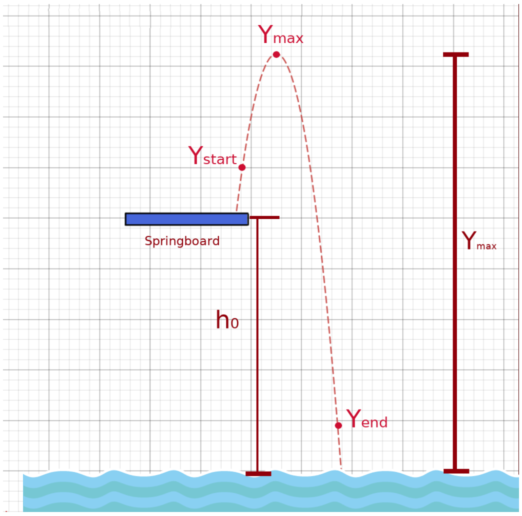


FIGURE 6. Diving motion peak height estimation.

The temporal encoder as shown in Figure 5 is inspired by [4] and comprises of 3 TCN blocks, each having a residual connection to the previous block. Each TCN block contains two sets of 1D convolutional layer, group normalization layers, ReLUs for non-linearity and dropout layers with a probability of 0.5 for regularization. The concatenated feature vectors from the backbone network are input to the temporal encoder. A sequence of length T $\{\phi_1, \phi_2, \ldots, \phi_\mathrm{T}\}$, where ϕ_t is the feature vector of the t^{th} frame, is fed into the encoder. The output of the proposed network is a single vector $\hat{o}_\mathrm{t}$ for each temporal window as shown in equation (2). The target vector y is one hot encoding, which has the class label one for positive classes and zero for negative classes. The weights vector w has value w_c for the respective c class. We set the weights vector w_c empirically to [0.15, 0.35, 0.15, 0.35] (see section IX-D for more details). We use a weighted cross entropy loss function as presented in equation, for training, where C is the number of output classes:

$$\mathcal{L} = -\sum_{i=1}^{C} w_i \cdot y_i \cdot log(\hat{o}_t)), \tag{1}$$

$$\hat{o}_t = softmax(f_{TCN}(\phi_1, \phi_2, \ldots .\phi_T)). \tag{2}$$

IV. PEAK DIVE HEIGHT ESTIMATION

The dive peak height estimation uses the localized action time computed using the action localization TCN Network. The Figure 6 shows typical COM trajectory and its peak height Y_{max}. The total time T_{dive} of the diving motion is computed using the equation (3). The F_{end} and F_{start} represents the frame numbers in the untrimmed video. The V_{freq} is the recorded video frequency of the untrimmed video, they relate as follows:

$$\Delta T_{dive} = \frac{(F_{end} - F_{start})}{V_{freq}} \tag{3}$$

The highest point of the trajectory Y_{max} is computed as:

$$Y_{max} = Y_{start} + \frac{1}{2g}\left(\frac{Y_{end} - Y_{start}}{\Delta T_{Dive}} + \frac{g\Delta T_{Dive}}{2}\right)^2, \tag{4}$$

based on [18].

The Y_{start} is the vertical position of the body COM represented in meters at the start of the dive relative to the water-level. Y_{end} represents the body COM position at the end of the dive above water. And g is the gravitational constant. The assumed values for $Y_{end} = 1$ and $Y_{start} = Y_{start} = h_0 + 1$. The height h_0 is the springboard height above the water and is assumed to be known for a given diving video, e.g. $h_0 = 3$ for a three meter tower. Typically, we assume the body COM for an average athlete as $1m$ above the springboard. Note that in case the exact measurements of height and body mass of the diver are available, then a more precise COM position can be used instead of assumed values.

The following section provides the mathematical derivation for the Equation 4.

A. MATHEMATICAL DERIVATION

We use the following Kinematics equation for projectile motion, where a position of the COM of an object is described with $p(t) = [X(t), Y(t)]^T$ as a point in a 2D plane and its corresponding velocity $v(t) = [U(t), V(t)]^T$. Since the vertical (Y-axis) component only contributes to the height, we consider only the y-component of the COM position and velocity.

$$Y(t) = Y_{start} + V_{start} + \frac{1}{2}at^2 \tag{5}$$

$$V(t) = V_{start} + at \tag{6}$$

Furthermore, we are only interested in the trajectory of the COM in the air, the only acceleration acting on the COM is gravity, i.e. $a = -g$. Here, t_0 is the initial time and $Y_{start} = Y(0), V_{start} = V(0)$ are the initial position and velocity in y-direction, respectively. The highest point (Y_{max}) can now be computed as follows. We know, the velocity at this point in time (t_{max}) is 0 i.e. ($V(t_{max}) = 0$).
Using equation (6) we can compute this time via:

$$\begin{aligned} 0 &= V_{start} - gt_{max} \\ \Rightarrow t_{max} &= \frac{V_{start}}{g} \end{aligned} \tag{7}$$

Using equation (5 and 7) we obtain:

$$\begin{aligned} Y_{max} &= Y_{start} + V_{start}t_{max} - \frac{1}{2}gt_{max}^2 \\ &= Y_{start} + V_{start}\frac{V_{start}}{g} - \frac{1}{2}g\left(\frac{V_{start}}{g}\right)^2 \\ &= Y_{start} + \frac{(V_{start})^2}{2g} \end{aligned}$$

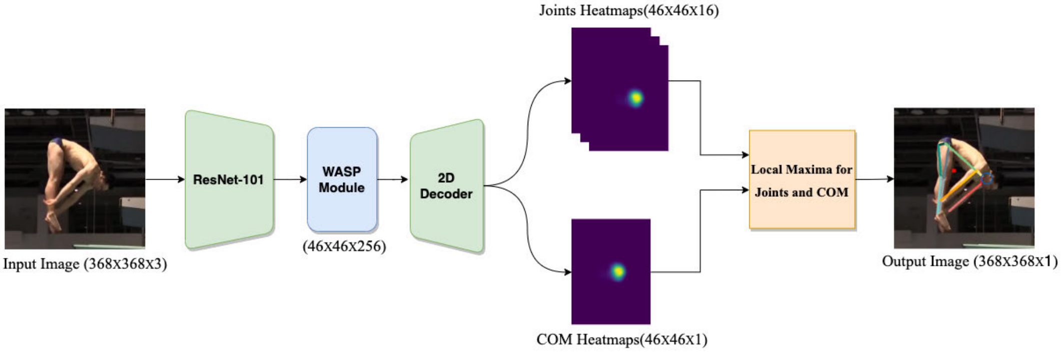


FIGURE 7. **The architecture of DiveNet pose regression model. The input image(HxW) is fed through the ResNet backbone and Waterfall Atrous Spatial Pyramid(WASP) module to obtain 256 feature channels at reduced resolution by a factor of 8. The decoder module generates 17 heatmaps, one per 16 joints and one for COM, followed by a local max operation which produces 16 keypoints and 1 COM (as shown in output image with a red circle).**

Now, utilizing (6) we can write: $V_{start} = V(t) + gt$. We can now approximate V_{start} via an average approximation as:

$$\begin{aligned} V_{start} &\approx V(\frac{\Delta T_{Dive}}{2}) + g\frac{\Delta T_{Dive}}{2} \\ &\approx \frac{\Delta Y}{\Delta T_{Dive}} + g\frac{\Delta T_{Dive}}{2} \\ &\approx \frac{Y_{start} - Y_{end}}{\Delta T_{Dive}} + g\frac{\Delta T_{Dive}}{2}. \end{aligned} \tag{8}$$

Inserting this into equation (8) gives:

$$Y_{max} = Y_{start} + \frac{1}{2g}\left(\frac{Y_{start} - Y_{end}}{\Delta T_{Dive}} + \frac{g\Delta T_{Dive}}{2}\right)^2, \tag{9}$$

as required.

V. DIVER POSE

This following section describes the pose regression module of the DiveNet architecture.

A. POSE REGRESSION NETWORK

The Figure 7 shows the DiveNet Pose regression network architecture which uses a ResNet-101 backbone feature extractor network. A Waterfall Atrous Spatial Pyramid(WASP) module inspired by [48] replaces the final layers. The decoder receives 256 low level feature maps from the first block of the ResNet backbone and 256 feature maps from WASP. The feature maps are aggregated and processed through convolutional layers, dropout layers ($p = 0.5$), and a final bilinear interpolation to scale to the original input size after a max pooling operation to match the dimensions of the inputs. We change the final decoder layers to generate $K + 1$ heatmaps where K is the number joints along with single heatmaps for the COM. The probability distributions were obtained by applying a softmax function. We train DiveNet pose regression network to regress 2D diving pose along with COM. As a criterion, the mean squared error (squared L2 norm) was used between ground the true heatmaps (H_{x_i}) and predicted heatmaps (H_{y_i}),

$$Loss_{mse} = \frac{1}{n}\sum_{i=1}^{n}(H_{x_i} - H_{y_i})^2, \tag{10}$$

where, $H_{x_i} = [h_1, h_2, .., h_{16}, cx_i]$ and $H_{y_i} = [h_1, h_2, .., h_{16}, cy_i]$. Here, $h_1, h_2, .., h_{16}$ are the heatmaps of joints and cx_i and cy_i are heatmaps of the prediction and ground truth COM, respectively.

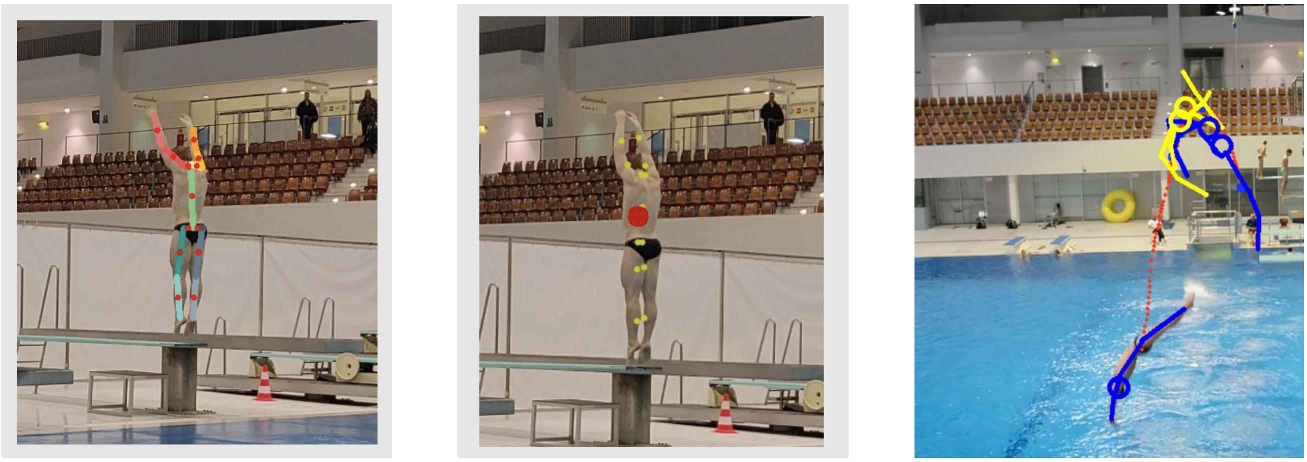
B. CENTER OF MASS TRAJECTORY

We compute the COM trajectory as illustrated in Figure 8a. First, we compute the midpoints of bone segment representing bone centroids (a). Next, we compute a weighted sum of centroids to represent the COM (see Figure 8b). In order to compute the COM, we used a weighted sum of bone segment centroids as proposed in [58], [59], and [60]. Finally, we compute the COM on each frame to obtain the resulting trajectory in the image space (see Figure 8c).

VI. PROPOSED DIVING PLANE HOMOGRAPHY ESTIMATION

In this section, we derive how we estimate a homography to identify points from the 2D image plane from unknown camera view to a physical 2D plane. A homography provides a projective geometry of two cameras and a world plane. In simple terms, homography maps images of points which lie on a world plane from one camera view to another. However, our setup consists of a single camera and without any known camera calibration procedure or precise depth information available about diving plane.

We begin with the motion assumptions from [18] for water diving. The primary assumption is that the divers COM follows a projectile motion while the diver is airborne. Thus, the COM can be completely described with a 2D parabola in a physical 2D plane. We aim to register the physical 2D plane (looking from an orthogonal point of view onto the plane) with the camera plane (might have a non-orthogonal view angle) via the homography estimation. The Figure 9 shows the unknown recording image view plane, physical



(a) Segment midpoints (b) COM computation (c) COM trajectory

FIGURE 8. Illustration of the COM computation. First, we compute the midpoints of bone segments representing bone centroids (a). Next, we compute the COM by a weighted sum of centroids (b). The COM trajectory in the image space (c).

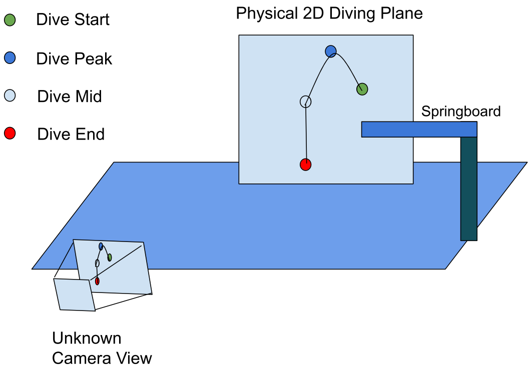


FIGURE 9. Diving motion COM trajectory estimation using homography.

diving plane and selected four center of mass trajectory points for registration. The mathematical derivation to approximate physical peak heights derivation is explained in section IV-A. The homography

$$H = \begin{bmatrix} h_{11}, h_{12}, h_{13} \\ h_{21}, h_{22}, h_{23} \\ h_{31}, h_{32}, h_{33} \end{bmatrix}$$

maps the image coordinates (x, y) to the respective physical plane (x', y'), i.e.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \sim H \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \tag{11}$$

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \tag{12}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}. \tag{13}$$

Given 4 points, a homography could be solved with a unity constraint to cover the last degree of freedom, i.e.

$$Ah = 0, \tag{14}$$

$$s.t. \quad \|h\|_2 = 1, \tag{15}$$

with $h = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T$ and A can be derived by rearranging (12), (13).

Via the COM tracking from section V-B, we can easily identify multiple points in the image plane ($(x, y)_i$) that are not lying on one line. The challenge in the present setting is that we can only estimate the respective physical points in the y-direction, via the equation (5) and the respective time-stamp of the dive. However, we do not have any information about the x-direction, since no starting angle is given that could be used to estimate this component via a starting velocity.

In this work, we propose to estimate a coefficients of a quadratic function jointly with the homography that defines the x coordinates, together with a relation on certain points that have to hold via the Cayley-Klein metric, defined via the cross-ratio.

To obtain a minimal number of parameters to estimate, we describe the projectile motion in the physical plane, via a parabola centered at the highest point $a_1 = Y_{max}$, with the restriction that we only allow concave functions, i.e.

$$y(t) = -a_0 x(t)^2 + Y_{max}, \tag{16}$$

$$s.t. \quad a_0 > 0. \tag{17}$$

This can also be formulated as unconstrained equation

$$y(t) = -\exp(\tilde{a}_0)x(t)^2 + Y_{max}, \tag{18}$$

with $\tilde{a}_0 = \log a_0$. The $x(t)-$ coordinate for the respective point $y(t)$ on the parabola can be described as

$$x(t) = \begin{cases} -\sqrt{\dfrac{-y(t)+Y_{max}}{\exp(\tilde{a}_0)}} & : t > t_{max} \\ \sqrt{\dfrac{-y(t)+Y_{max}}{\exp(\tilde{a}_0)}} & : t \leq t_{max} \end{cases} , \tag{19}$$

with t_{max} given via equation (7). With (19), (18) and we can describe coordinates (x', y') on the physical plane for different instances in time. However, to do this we still need to estimate $\tilde{a}_0$, which leads to, at least, one condition for the x-direction on the physical plane. To this end, we create a ratio that has to match between to two coordinate systems. Let us start by computing the time (t_f) that the COM needs to fall from the highest point (Y_{max}) at time t_{max} until the y position is back to the same height as the initial height (Y_{start}). Looking at (7) at $t = t_{max}$, we have $V_{start} = 0$ and $Y_{start} = Y_{max}$, thus the time until the initial height is reached again, can be computed by rearranging the equation

$$Y_{start} = Y_{max} - \frac{gt_f^2}{2} \tag{20}$$

$$\Rightarrow t_f = \sqrt{\frac{2(Y_{max} - Y_{start})}{g}}. \tag{21}$$

Thus the total time from take-off until the same height is reached again is $t_r = t_{max} + t_f$. The time span t_r describes the amount of time the diver needs to jump, pass through the highest point and come back to the same height. Together with $t_0 = 0$ and t_{max}, the respective points $(p(t))$ of the COM trajectory in the image plane, we can span a triangle in the 2D plane, were the divers COM lies in. We use this fact to create a relation between a line pointing along the x direction (in the plane) and the y direction (in the plane) defined as follows

$$d_x^{\text{img}} = p(t_r) - p(t_0) \tag{22}$$

$$p^{avg} = p(t_0) + \frac{1}{2}(d_x^{\text{img}}) \tag{23}$$

$$d_y^{\text{img}} = Y_{max} - (p^a vg). \tag{24}$$

To obtain the relation, we exploit the Cayley-Klein metric, which is based on the cross ratio that is invariant to perspective changes. The metric measures the distance between two points a and b, denoted as $|ab|$, with respect to a quadric, e.g. a circle, with intersection points q and p; it is defined as

$$d(a, b) = C \log(\frac{|bp||qa|}{|ap||qb|}), \tag{25}$$

for a constant C. Since we have two distances (x direction and y direction) that we compare we define one quadric for both distances, defined to be the circle with center p^{avg} and radius $r = \|d_y^{\text{img}}\|_2$. The circle can thus be parametrized as

$$x(f) = p^{avg} + r\cos(f) \tag{26}$$

$$y(f) = p^{avg} + r\sin(f) \tag{27}$$

The points p and q for the respective directions (x and y, thus denoted as p_x, q_x and p_y, q_y respectively) can now be computed via the intersection point f_x^* and f_y^* of the following lines with the circle

$$p_x = p^{avg} + f_x^* d_x^{\text{img}} \tag{28}$$

$$q_x = p^{avg} - f_x^* d_x^{\text{img}} \tag{29}$$

$$p_y = p^{avg} + f_y^* d_y^{\text{img}} \tag{30}$$

$$q_y = p^{avg} - f_y^* d_y^{\text{img}}. \tag{31}$$

The intersections can thus be computed to be

$$f_x^* = \sqrt{\frac{r^2}{\|d_x^{\text{img}}\|_2^2}} \tag{32}$$

$$f_y^* = \sqrt{\frac{r^2}{\|d_y^{\text{img}}\|_2^2}}. \tag{33}$$

Now we can compute the relation between x and y direction via a quotient of the following distances

$$rel_{xy} = \frac{d(p(t_r), p^{avg})}{d(Y_{max}, p^{avg})}. \tag{34}$$

Note that the constant C cancels out, since we use the same quadric for both distances.

This finally gives a relation for the x direction in the physical plane that we can use in the estimation of the homography with included parabola constraint to obtain the remaining degree of freedom for the parabola. The distance in x direction, in the physical plane can thus be computed from the velocity in y direction, multiplied with the ratio and the time from the time span t_f, as

$$x^{phy}(t_f) = \frac{-(Y_{max} - Y_{start})}{t_f} rel_{xy} t_f \tag{35}$$

$$= -rel_{xy}(Y_{max} - Y_{start}). \tag{36}$$

The final optimization problem to solve reads

$$\min_{\tilde{h}} \|A\tilde{h}\|_2 \tag{37}$$

$$s.t. \quad \|h\|_2 = 1 \wedge x^{phy}(t_f) = x(t_f), \tag{38}$$

were $\tilde{h} = [h^T, \tilde{a}_0]^T$.

VII. IMPLEMENTATION DETAILS

In this section, we provide details about the diving datasets and the training procedure of the dive-net.

A. DIVING DATASET

1) DSV DATASET

We use a recorded DSV diving dataset consisting of 450 diving sequences covering 15 hours of video data, with each motion clip lasting approximately 2 minutes. The video differs in varying camera viewpoints and springboard heights. The dataset consists of diving actions at four different levels of diving height: 3m, 5m, 7.5m, 10 meters. The diving action within a single video sequence ranges from 2 to 5 seconds, making the localization problem much harder. The dataset annotations consist of two temporal boundary marker frames denoting the beginning and end of the diving action. The annotations also consist of 2D keypoints representing 16 joints. Note, the recorded dataset videos are with a static camera with a frame rate of 60Hz and a resolution of 3840 × 2160 px. To the best of our knowledge, this is the first publicly available dataset with a static camera setup for diving sports. This enables us to compute homography to the diving plane via our newly proposed homography estimation method.

We use a train, validation and test split of 60%, 15% and 25%, respectively.

2) IAT DATASET

IAT dataset consists of 100 Olympic diving performance television broadcasting videos performed over 3 meter high springboard. The video differs in varying camera viewpoints. The dataset annotations consist of two temporal markers to notify the beginning and end of the diving action. The dataset annotations consist of two temporal boundary marker frames denoting the beginning and end of the diving action. The annotations also consist of 2D keypoints representing 16 joints. The videos involve a moving camera and is recorded with a frame rate of 25Hz and a resolution of 720 × 576 px.

3) SPORTSCAP

The dataset includes per-frame action labels, manually annotated poses, and action evaluations from professional referees of various challenging sports video clips [8]. The dataset separates sports activity into two categories: competitive sports and daily exercises. Competitive sports comprise balance beam, competitive diving, uneven bars, vault-women, hurdling, pole vault, and high jump, while daily exercises include boxing, keep-fit, and badminton. The dataset consists of 640 videos (110K frames) with 450,000 annotated skeletons of 25 joints and accompanying bounding boxes. We trained our model using only diving sports 2d pose data, totaling 23635 frames. Although, the frames in the dataset are annotated for different video sequence, the annotations are only available for intermittent frames. Hence, we use the dataset for training purposes only and do not use it for evaluations.

B. EXPERIMENTAL SETUP

The images used for training and evaluation are sampled uniformly from untrimmed diving videos at a rate of 60 frames per second. For training, we only use the frames involving diving motion. The images are cropped to size 224 × 224 around the bounding box containing the diver. The cropped images are converted into features on per frame basis using one of the feature extractors described in section III. The model is trained with temporal sequences of length T=17. The frames of a diving sequence are split into temporal windows of length 17. These set of features are used as inputs to the temporal encoder network.

C. TRAINING

We use TCN temporal encoder to have a receptive field of 17. The 1D convolution layers have kernels size of 3 and padding size of 2. The dilation size is set to 1. We train all the networks for 50 epochs with a batch size of 16. Adam Optimiser [61] is used for the optimisation of the networks. The dropout rate is set to 0.5. We use three temporal blocks. The learning rate and weight decay are set to 0.00001 and 0.001, respectively. The weights used for cross-entropy loss are [.15,.35,.15,.35]. The model is implemented in Python 3.6 using PyTorch on NVIDIA GeForce GTX 1080 GPU. We train DSV diving dataset with the following split: 300 sequences for training, 50 sequences for validation and 100 sequences for testing. The frames in each sequence belong to one of the four classes (No Dive, Start, Mid, End). We train the DiveNet-pose regression model for 50 epochs. Adam was used as an optimizer with 0.0001 learning rate. The accuracy stabilized at around 17 epochs with batch size of 8. We train the model on DSV, IAT and Sportscap [8] Dataset.

VIII. EVALUATION RESULTS

A. ACTION LOCALIZATION

This section discusses the result of various experiments we performed to determine the optimal parameters for the action localization model for diving. An example of ground truth and predicted output probabilities is shown in Figure 10. For the given video sequence, the start and end markers are predicted within a single frame latency with respect to target temporal markers. The predicted pink and purple coloured area shows the latency of start and end boundary markers respectively over the bold lines representing the ground truth.

We evaluate our diving action localization model with two metrics: (i) classification accuracy of the frame and (ii) latency in the position of boundary markers.

a: CLASSIFICATION ACCURACY

The mean classification accuracy is the ratio of the number of correctly classified frames to the total number of input frames representing boundary classes. Due to the presence of highly imbalanced classes (start and end class frames having very low representation), mean accuracy can be misleading and incorrect metric. To this end, we further analyze the performance of a model with a confusion matrix as shown in Figure 12. The confusion matrix clearly shows a low number of misclassification for all the classes, which is essential for accurate localization of the diving motion.

b: LOCALIZATION LATENCY

We also evaluate the latency of the predicted boundary markers of the trained model. Latency gives additional information about how far are the start and end markers predicted from the ground truth in time. For example, given a

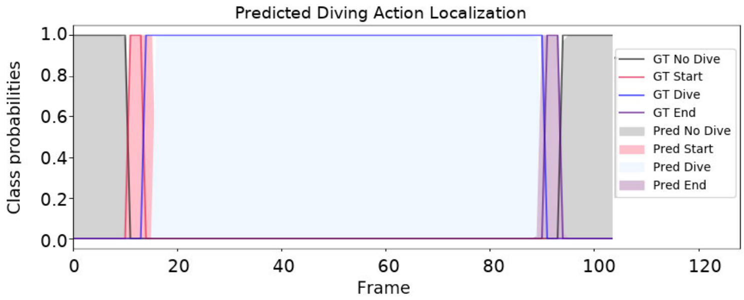


FIGURE 10. The ground truth and predicted probability signals for the 4 class model (No Dive, Start, Dive, End). Bold lines represent the ground truth probability signal, and coloured areas represent the predicted probabilities.

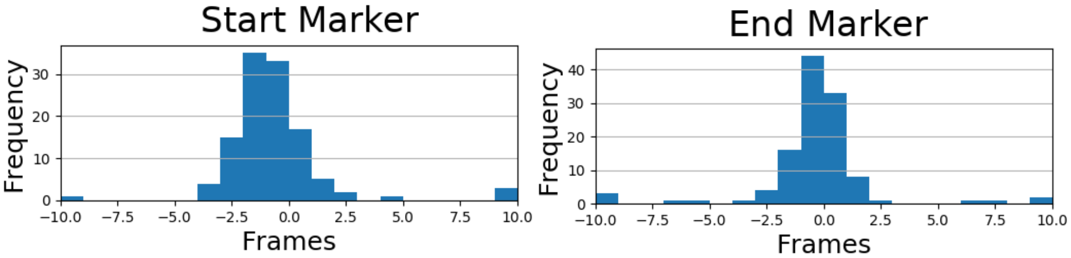


FIGURE 11. Histogram of the latency measure of the estimated start and end boundary markers. The y-axis represents the number of sequences and x-axis represents the lag in frames. The negative values on x-axis represents prediction prior to the reference frame and positive values represents prediction after the reference frame.

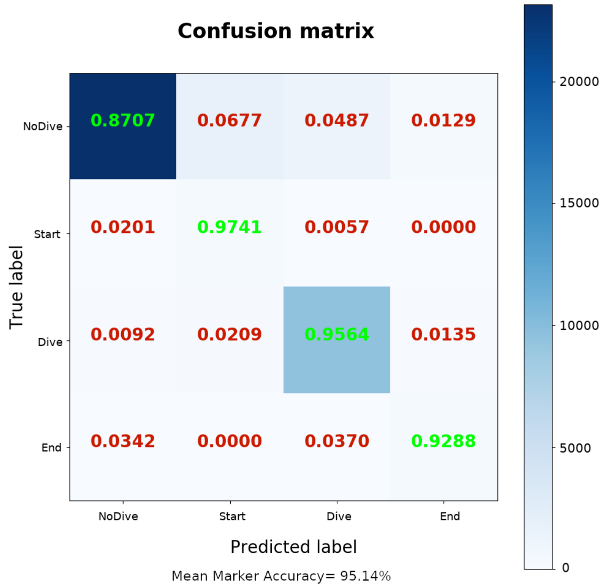


FIGURE 12. The confusion matrix shows the model accuracy over test set. The model uses a HMR backbone network with 17 frames as input.

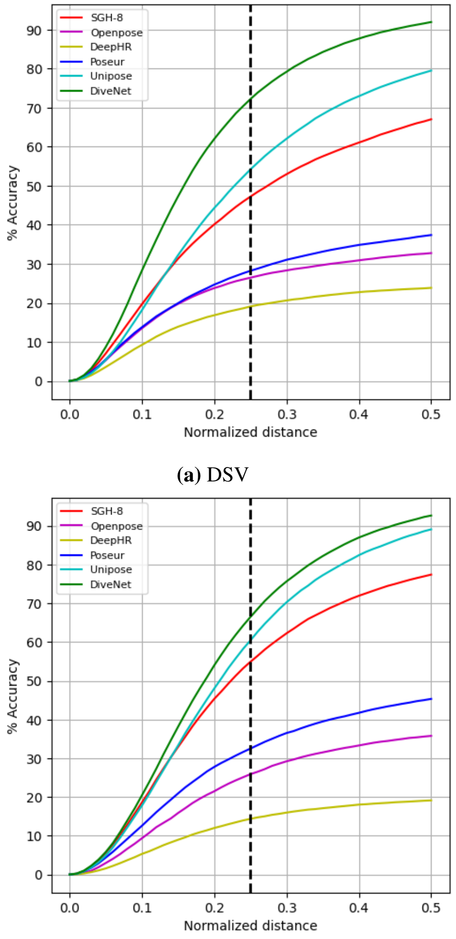
sequence where the start marker is the i^{th} frame of the video, and the network predicts the j^{th} frame to be the start marker, the sequence is termed as accurately detection if the latency $|(j-i)|$ lies within a threshold limit of 1. We measure the percentage of test sequences that are within the low latency threshold limit. The higher the percentage over the test set, the better the predictions (see Figure 11).

TABLE 1. The PCK@0.25 accuracy metric showing the accuracy of pose regression models on DSV and IAT test dataset.

	DSV Dataset					IAT
Model	**3m**	**5m**	**7.5m**	**10m**	**Mean**	**3m**
SHG-8 [44]	56.44	44.48	31.88	31.69	41.12	58.81
Openpose [47]	35.81	17.66	14.04	10.18	19.42	28.49
DeepHR-Net [46]	34.69	5.165	2.25	1.44	10.88	13.87
Poseur [51]	42.76	16.3	14.68	9.09	20.71	33.72
UniPose [48]	61.5	49.87	40.56	48.125	50.01	61.30
Unipose-FT	77.98	**74.91**	70.28	68.05	72.80	**70.43**
DiveNet	**78.46**	74.37	**72.04**	**70.28**	**73.78**	67.36

B. 2D POSE REGRESSION

We evaluate our 2D pose regressor in comparision to state-of-the-art 2D human pose regression models on diving-sports datasets. We use Percentage of Correct Keypoints normalized using headlength (PCKh). The predicted key point is assumed correct when it lies within a certain normalized threshold distance of the head length. For example, PCKh@0.25 metric is the percentage of correct keypoints at a threshold of 0.25% times of the head length. We found different threshold levels (typically 0.5, and 0.2) of PCK-h accuracy reported in literature [10], [11]. Moreover, we observed empirically that the threshold of 0.25 represents the joint area in pixels in the test set images reasonably. Hence, we use 0.25 as one representative threshold of PCK-h metric, in Table 1. The performance on other thresholds can be observed in Figure 13, which shows the diverging accuracy performances on the diving test dataset (DSV - (a), IAT- (b)) for various models at different threshold levels (x-axis represents the different thresholds). The figure clearly shows that the



(a) DSV

(b) IAT

FIGURE 13. The PCK-h plot compares the accuracy of different models at different PCK-h thresholds. The plot (a) shows the performance on DSV test dataset and plot (b) shows the performance on IAT test dataset.

DiveNet Model gives superior performance for different thresholds by achieving over 90% accuracy on both datasets at the normalized threshold distance of 0.5.

The pose regression accuracy scores are measured in PCK and can be found in Table 1. We found that Unipose performs best in all the state-of-the-art models without any training on a diving dataset. Hence, we further finetuned the unipose model (named Unipose-FT) on the diving dataset and obtained an accuracy of 72.80% on DSV and 70.43% on IAT dataset. DiveNet achieved a slightly higher PCK@0.25 accuracy of 73.78% on DSV dataset but a slightly lower accuracy of 67.36% on the IAT dataset.

Note that beside the competitive performance of DiveNet regarding pose estimation, the approach provides increased accuracy compared to the other approaches in estimating physical parameters, as discussed in section VIII-C

Figure 14 depicts scenarios where DiveNet outperforms other models in terms of generating more realistic and precise joint locations. We can observe in most cases that the models were unable to correctly identify the upper body, resulting in a poor head position. In first row, SGH-8, OpenPose and Poseur failed to detect and predicted the left arm completely. The left elbow was mistaken for the left knee by DeepHRNet. OpenPose and DeepHRNet failed completely for the head. Due to its inability to detect both ankles, Unipose gave lesser lengths for the legs. While DiveNet was able to achieve better outcomes for the hands, legs, and head. The images in the second row have background glare from lights, making the scenario more difficult.The SGH-8 incorrectly predicted hip and shoulder as left foot, resulting in incorrect head prediction. Only the left hand could be reliably predicted by OpenPose, while Poseur otherwise failed to detect any joint. Right hand and right leg were combined with left leg via DeepHRNet. Unipose confused right hand with background. While it is clear that DiveNet offers a better answer in this case. The third row shows an example with a blurred image of a diver about to complete a dive. Due to the blurry image in the hip area, none of the models were able to accurately estimate the left hip joint. SHG-8 was unable to detect the left leg due to occlusion. OpenPose failed miserably and predicted the right leg as the left hand. Poseur failed to detect the upper part of body. DeepHRNet projected legs as hands because it believed the individual to be standing upright. Unipose outperforms other models in the hands, legs, and shoulders, while DiveNet outperforms Unipose in the whole skeleton, notably in the hands, hips, shoulders, and head. Similar to the second row, the pose in the fourth row is challenging due to a high left body occlusion. This scenario involves a position in which the hands go around the legs, causing SHG-8 and DeepHRNet to interpret the left hand as the right leg. OpenPose was unable to identify the upper body, and Unipose failed to predict the left elbow correctly. We can see that, despite the difficult circumstances, DiveNet's prediction was fairly accurate. In the fifth row, SHG-8 failed for the right leg and OpenPose failed for the left leg. Additionally, SHG-8 has poor predictions for hands. DeepHRNet failed for the right hand, while Unipose failed for both hands and the right ankle. When compared to all other models, DiveNet provides more accurate hands, legs, and shoulder position. The sixth, seventh, and eighth rows all include a situation with additional individuals seated in the background. We can observe in OpenPose, Poseur and DeepHRNet in row six, and DeepHRNet in rows seven and eight, that they failed in this case, and anticipated a diver from the background. SHG-8 failed in rows six and eight for hip joints and right hand respectively. In rows seven and eight, OpenPose failed for the right hand. Unipose outperformed other models, although DiveNet's results were more precise and accurate, particularly for the hips, shoulders, and legs.

Figure 15 depicts the extreme conditions under which pose regression models failed for various body joints. This figure also depicts the limitation scenarios where DiveNet predictions were less accurate. The first row shows the complex pose which are difficult to estimate. The SHG-8

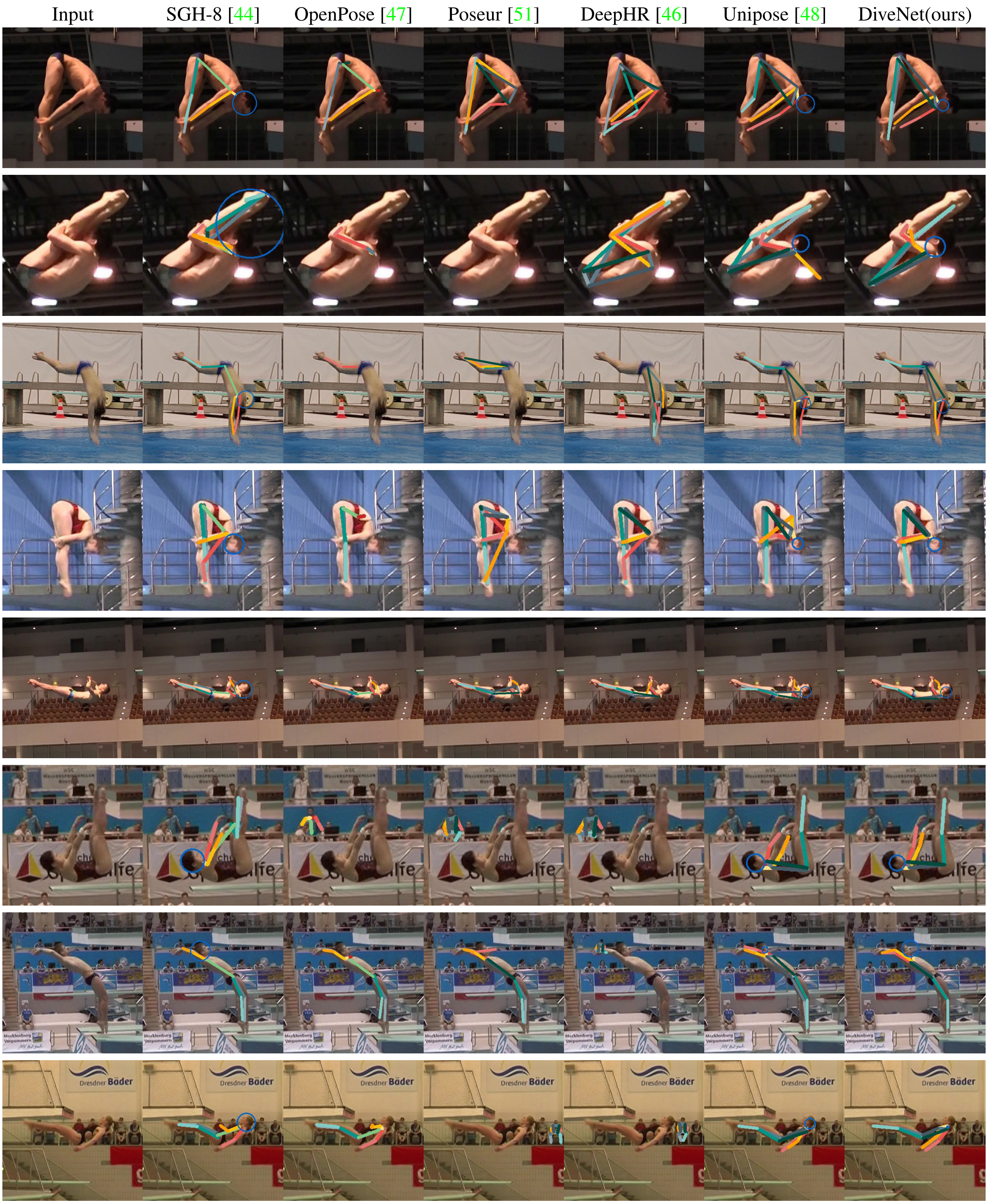


FIGURE 14. The figures demonstrate the 2D pose estimate results of various models on the DSV Diving set. Divenet shows superior pose regression results on challenging diving images.

and DeepHRNet messed up the hands and legs completely. As a result, SHG-8 misjudged the position of the shoulder, mistaking the hip for the position of the head. Only the left knee and ankle were predicted by OpenPose, and they were confused with the right ones. Unipose mistook the left hand for the right one when making its prediction. DiveNet offers a more effective solution, although it fails to accurately predict legs. Due to the background, none of the models were able to estimate the hands correctly in the second or third row. DiveNet's predictions for the second row may be seen with

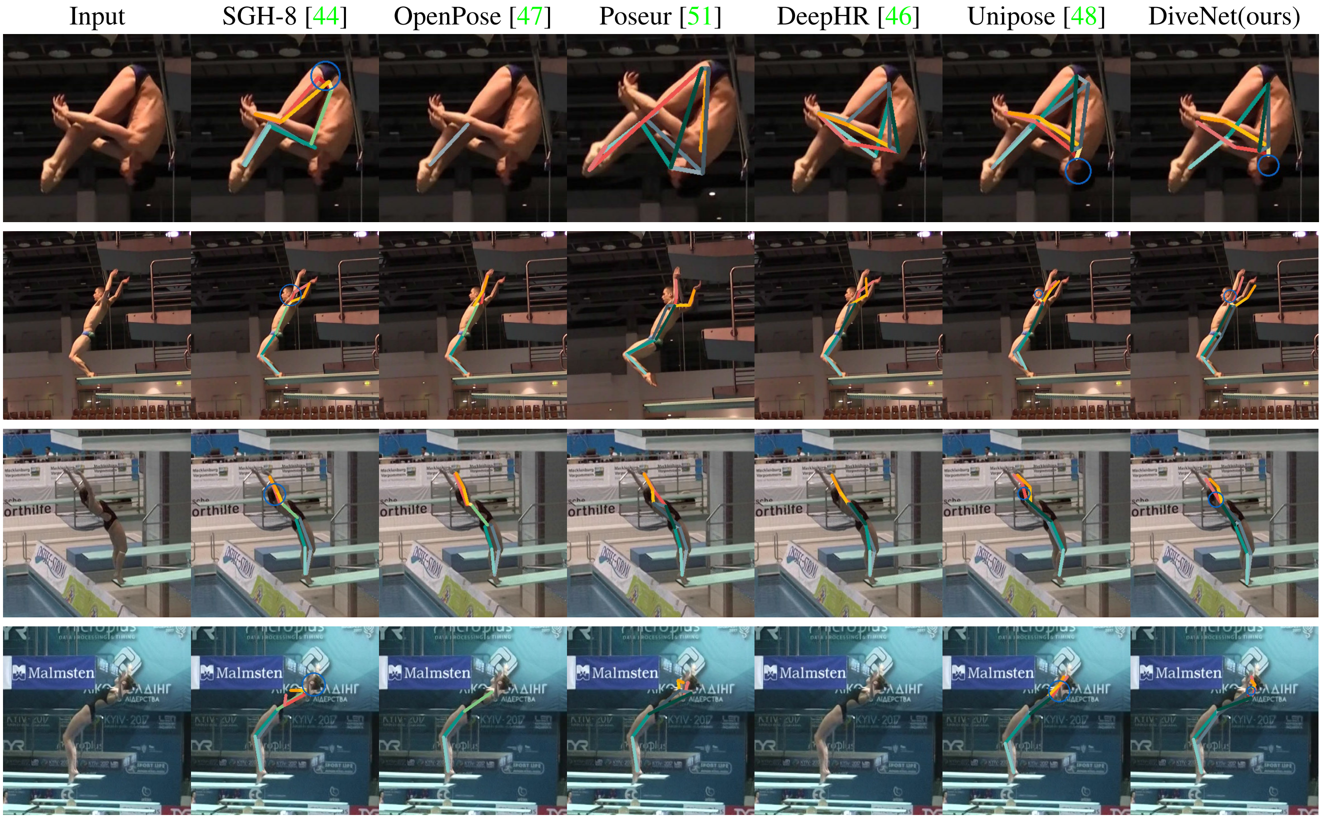


FIGURE 15. The figures demonstrate the predictions of various models on exceedingly challenging scenarios in the DSV and IAT Diving sets. The last column shows the current limitations of DiveNet's predictions on challeging scenarios.

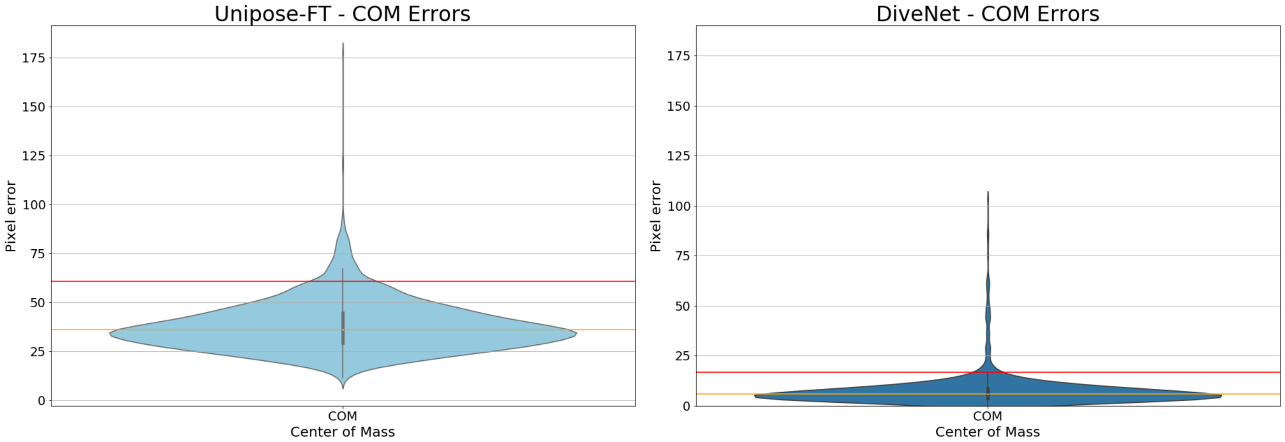


FIGURE 16. The plots show the COM error distribution of Unipose-FT (left) and DiveNet (right) model. The median COM error (left-orange line) of Unipose-FT model is 36.08 pixels and 95 percentile error threshold (left-red line) lies at 60.76 pixels. The median COM error of DiveNet (right-orange line) is significantly lower, at 5.87 pixels and the 95 percentile threshold (right-red line) lies at 16.64 pixels.

some offsets for the hands and right side of the body. In the third row, Unipose and DiveNet incorrectly identified the left leg as the right and struggled to identify the left hand. The last row illustrates the situation where the upper body area around the face is indistinct and has background noise. The skin tone seems to hide the hands and face, making them hardly apparent. We can see that DeepHRNet was ineffective in predicting any joint positions. All of the models, including DiveNet, failed to predict hands.

C. PHYSICAL PARAMETER ACCURACY

We measure different physical parameters of diving performance: COM, Hip Angle and Knee angle. Also, we only compare DiveNet with Unipose-FT instead of Unipose Model inorder to have fair comparison of models with same training setup and regressing a same number of joints. The Table 2 shows the accuracy of the COM prediction over the test set. The median COM prediction error is 5.87 pixels on DSV and 6.05 pixels on IAT dataset. This is significantly

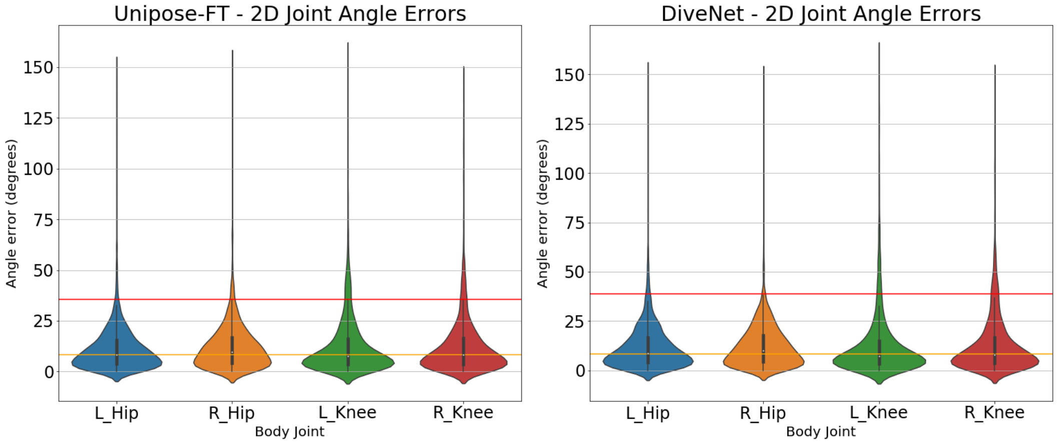


FIGURE 17. The plots show 2D joint angle error distribution of Hip Angles and Knee Angles for Unipose-FT (left) and DiveNet (right). The median angle error (left-orange line) of Unipose-FT model is 8.28 degrees and 95 percentile error threshold (left-red line) lies at 35.435 degrees. The median angle error of DiveNet (right-orange line) is marginally higher at 8.52 degrees and the 95 percentile threshold (right-red line) lies at 38.96 degrees.

TABLE 2. The table shows the COM physical parameter mean pixel errors over different height jumps performed in the test set of diving datasets.

	DSV Dataset					IAT
Dive Height	**3m**	**5m**	**7.5m**	**10m**	**Mean**	**3m**
UniposeFT	43.96	31.16	30.04	32.05	34.30	37
DiveNet	**6.26**	**5.44**	**5.23**	**5.23**	**5.54**	**6.05**

TABLE 3. The table shows a comparison of the mean joint angle errors in degrees over different height jumps in the diving test set. The lower the angle value, the better is the model performance. We observed less than 1 degree difference and no significant variation in Hip angle and Knee angle error between the two models.

		DSV Dataset					IAT
	Dive Height	**3m**	**5m**	**7.5m**	**10m**	**Mean**	**3m**
Unipose-FT	Hip Angle	7.96	10.35	9.12	9.71	**9.28**	8.51
	Knee Angle	8.09	9.35	13.13	8.88	9.86	**6.03**
DiveNet	Hip Angle	7.77	12.19	11.70	9.91	10.39	**8.20**
	Knee Angle	7.90	8.63	14.56	7.82	**9.73**	6.30

TABLE 4. The table shows the sensitivity in peak height estimation of the diving trajectory measured in meters over different height jumps performed in the test set of DSV diving datasets.

	DSV Dataset					IAT
Dive Height	**3m**	**5m**	**7.5m**	**10m**	**Mean**	**3m**
Peak height	±0.19	±0.13	±0.13	±0.13	±0.14	±0.21

lower than Unipose-FT model which results in 34.30 and 37.00 pixel on DSV and IAT datasets respectively. Due to the lack of availability of actual transformation of images to the real diving trajectory plane, we restrict our evaluation to pixel space. The mean joint angle error are as shown in Table 3 for hip and knee joints are 10.39 degrees and 9.73 degrees respectively. The peak height estimation sensitivity is influenced by the action localization module. We observed a mean peak height error of 0.14 meters on DSV dataset and 0.21 meters on the IAT dataset as shown in Table 4. The Figure 16 shows the error distribution plot for COM (left) and joint angles (right plot). The COM error distribution shows 95 percentile of errors lie under 16.64 pixels for the complete diving testset. Similarly, the Figure 17 show the errors distribution plot of 2D angle errors of two trained models . The median angle error of Unipose-FT model is 8.2825 and 95 percentile error threshold lies at 35.435 degrees. The median angle error of DiveNet is marginal higher at 8.52 degrees and the 95 percentile threshold lies at 38.96 degrees.

IX. CONCLUSION

We present a diving motion analysis framework, DiveNet to extract performance parameters. A generic action localization (or recognition) of daily activities often performs well for most actions. However, the diving motion, which involves complex poses and manoeuvres, necessitates a specialized network to accurately localize and segment long untrimmed videos. We propose a diving action localization network, a novel temporal convolution network-based block that accurately localizes action with low latency using Spatio-temporal features over video frames. Our proposed method accurately predicts the start and end of the diving motion with more than 90% classification accuracy and latency accuracy of 90% within a threshold of $[-1, 1]$ marker positions. We also investigate the optimal context required to achieve high localization accuracy with low latency. In addition, we train a dive pose regression model, which outperforms all the 2D pose regression models on diving dataset. The trained pose regressor is robust for all diving heights and provided accurate poses even in challenging scenarios. The trained models are used to achieve a high accuracy on COM trajectory estimation around 6 pixels and 2D joint angle estimation accuracy around 10 degrees. We believe the presented physical parameter extraction can easily be generalized to other sports action tasks. Thus, we would extend the learning to other sports, especially sports with varying speeds of motion, as part of future work.

TABLE 5. Quantitative results: Ablation study results of different type of feature extractor with different input length. For instance, HMR_5 is Human Mesh Recovery model with five frames as input. The metric is per frame classification accuracy, the higher the better. The top two best results are highlighted in bold.

Backbone Network	Classes				Mean Accuracy
	No Dive	**Start**	**Dive**	**End**	
Resnet18_5	77.6	84	93.3	91.5	81.8
Resnet18_9	75.2	76.1	96.8	87.2	80.9
Resnet18_{13}	77.5	91.4	95.1	90.9	82.4
Resnet18_{17}	84.5	85.6	95.3	89.2	87.4
Resnet18_{21}	89.7	87.6	92.6	92.3	90.5
HMR_5	76.9	93.7	92.9	90.9	81.3
HMR_9	92.1	92	95.1	89.2	92.9
HMR_{13}	**95.2**	**93.4**	**95.2**	**89.7**	**95.1**
HMR_{17}	**91.5**	**94.5**	**96**	**90.6**	**92.7**
HMR_{21}	89.1	95.7	93.5	90.9	90.4

TABLE 6. Accuracy (*Acc*) and Latency (*Lat*) scores for models with different weighted configurations used in weighted cross entropy loss during the training. The latency values represent the percentage of frames that have latency between [−1, 1] in the respective columns.

Class Weights	Start Marker		End Marker	
	Acc	Lat(+/ − 1)	Acc	Lat(+/ − 1)
[.225, .272, .225, .272]	93.4	81	89.7	86.2
[.200, .300, .200, .300]	93.7	71.5	91.7	88.7
[.150, .350, .150, .350]	**94.5**	**90.5**	**90.6**	**87.9**
[.125, .375, .125, .375]	93.1	80.1	92	82.7
[.100, .400, .100, .400]	94.5	79.3	90.2	87.9
[.050, .450, .050, .450]	98.3	81.8	94.9	66.3

APPENDIX

ABLATION STUDIES

A. IMPORTANCE OF BALANCED SAMPLING

The diving dataset consists of temporal diving motion boundary markers denoted as a step function consisting of three frames (the boundary frame and adjacent +/ − 1 frames). With such a representation, the data is highly imbalanced with 95% of frames belonging to class "Dive" and "NoDive". The problem of imbalanced classes is, they lead to representation bias in classification results, leading to erroneous conclusions [7]. To this end, we use balanced sampling to sample an equal number of representatives for each class randomly at each epoch with weighted cross-entropy loss to improve the model performance.

B. BACKBONE FEATURE EXTRACTION NETWORK

We investigate two different backbone networks, as discussed in Section III for action localization task, see 5. We used ResNet-18 trained on Object classification task and HMR Network trained on a 3D Human Pose in the wild. The backbone network ResNet-18 [20] network and HMR network are trained with the same temporal input frame configurations {5, 9, 13, 17, 21}. The 3D human pose based feature extractor HMR provides superior results, reaching above the overall accuracy of 90%. The pose based extractor performs high despite slightly lower accuracy on the end marker.

The ResNet-18 based feature extractor performs poorly and does not reach above 90% accuracy for all four classes in all scenarios. Also, even when it achieves 91.4% accuracy for the "Start Marker", it fails for the "NoDive" class due to misclassification, resulting in high latency for "Start Marker". Furthermore, our experiments with deeper networks of ResNet architectures reported no improvements or inferior results.

C. NUMBER OF INPUT FRAMES

Table 5 presents experimental results with different input frames. The size of the temporal window fed to the temporal network as input affects the model's performance. We trained different models where N=5, 9, 13, 17, 21. The models with the input window sizes of $N = 5$ and $N = 9$ perform poorly. Thus, the model does not learn the necessary temporal embedding if the input window is too short. Hence, the model does not have adequate information to make accurate localizations.

On the other hand, too long sequences could also produce poor results, as the model is prone to overfit. With the larger temporal window of $N = 21$ frames, the model does not improve results. We found the input window size of $N = 13$ and $N = 17$ frames produced the best results. Therefore, we restrict further experiments and evaluation to the models with 13 and 17 frames input windows.

D. WEIGHTED CROSS ENTROPY LOSS

Balanced sampling does mitigate the issue of highly unbalanced data, but it does not provide the best results. Therefore, we further experiment with weighted cross-entropy loss. Even though [.20, .30, .20, .30] provides a stable accuracy for across the classes, its latency (Table 6) in predicting the marker position is poor, especially for the start marker (Figure 18). Thus, we found the weights of [.15, .35, .15, .35] provide, empirically, the best results of over 90% in all cases. The other class weight configurations deteriorate the latency of the boundary markers, which implies that they do not assist the learning of start and end boundary classes and are not suitable for the prediction model.

E. TEMPORAL CONTEXT

We experiment to analyze the importance of the bidirectional temporal context presented to the model. We trained two different models: (i) model with only past frames including the boundary frame F_t as input $[F_{t-n}, F_t]$ (ii) model with equal amount of past and future frames relative to the boundary marker frame $[F_{t-n}, F_{t+n}]$. We found the model trained with only the past frames ($[F_{t-n}, F_t]$) as input to predict the class of the current frame F_t under-performs with the accuracy of 70 %. Whereas by showing the information in both directions, i.e. frames immediately before and subsequent frames to boundary markers, the model prediction shows improvement over 91 %, as shown in Figure 19. Thus, it is clear that it is essential to have context from both directions to detect boundary classes accurately.

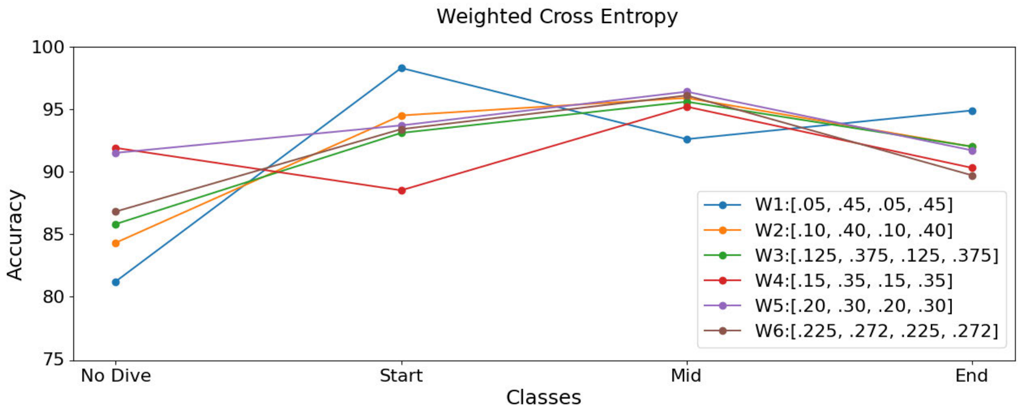
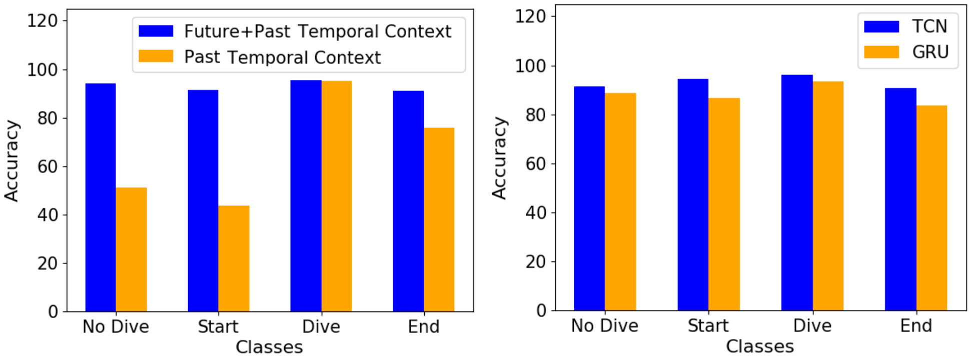


FIGURE 18. **Model accuracy over various individual classes using different class weighting configurations.**



(a) Impact on Accuracy of previous frames as input context vs bi-directional context for HMR-17 network.

(b) Accuracy results comparison of TCN as temporal encoder vs GRU [62] as temporal encoder with input size of 17 frames.

FIGURE 19. **Ablation study to analyze the required context in (a) and effectiveness of TCN with respect to GRU networks (b).**

F. TCN VS GRU

We experimented with two different sequence modelling architectures to evaluate diving action localization. We trained GRU architecture inspired by [62] and trained with the same experimental setup as our TCN architecture. The temporal encoders process features for bi-directional frames to classify a sample. The plots in Figure 19b clearly shows the superiority of TCN model over GRU [62] model. The TCN motion encoder outperforms the GRU model on all classes.

ACKNOWLEDGMENT

The authors would like to thank their colleagues at DSV and IAT for providing support in data gathering. They would also like to thank Dr. Thomas for his insights and comments during the course of work. They also would like to thank Ankit Dixit, Asmita Mittal, and Aditya Dash for their support in data annotations, preprocessing, and assisting in the feasibility study.

DECLARATIONS

- **Conflict of interest/Competing interests:** The authors declare that they have no conflicts of interest in this work.
- **Ethics approval:** Not applicable
- **Consent to participate:** Informed consent was obtained from all subjects involved in the study
- **Consent for publication:** Informed consent was obtained from all authors
- **Availability of data and materials:** The dataset link will be public for research purpose soon after the acceptance.
- **Code availability:** The parts of code can be made available on request.
- **Authors' contributions:** Conceptualization: Pramod Murthy and Bertram Taetz; methodology: Pramod Murthy and Bertram Taetz; software: Pramod Murthy; validation: Pramod Murthy and Arpit Lekhra; formal analysis: Pramod Murthy; investigation: Pramod Murthy and Bertram Taetz; resources: Didier Stricker; data curation: Pramod Murthy and Arpit Lekhra; writing—original draft preparation: Pramod Murthy; writing—review and editing: Pramod Murthy, Bertram Taetz, and Arpit Lekhra; visualization: Pramod Murthy; supervision: Bertram Taetz; project administration: Bertram Taetz and Didier Stricker; funding acquisition: Bertram Taetz and Didier Stricker. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] N. Kolotouros, G. Pavlakos, M. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2252–2261.

[2] T. T. Nguyen, V.-H. Le, D.-L. Duong, T.-C. Pham, and D. Le, "3D human pose estimation in Vietnamese traditional martial art videos," *J. Adv. Eng. Comput.*, vol. 3, no. 3, p. 471, Sep. 2019.

[3] W. Zhang, Z. Liu, L. Zhou, H. Leung, and A. B. Chan, "Martial arts, dancing and sports dataset: A challenging stereo and multi-view dataset for 3D human pose estimation," *Image Vis. Comput.*, vol. 61, pp. 22–39, May 2017.

[4] J. Zhang, P. Felsen, A. Kanazawa, and J. Malik, "Predicting 3D human dynamics from video," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7114–7123.

[5] A. Nibali, Z. He, S. Morgan, and D. Greenwood, "Extraction and classification of diving clips from continuous video footage," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 38–48.

[6] G. Kanojia, S. Kumawat, and S. Raman, "Attentive spatio-temporal representation learning for diving classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2467–2476.

[7] Y. Li, Y. Li, and N. Vasconcelos, "RESOUND: Towards action recognition without representation bias," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 513–528.

[8] X. Chen, A. Pang, W. Yang, Y. Ma, L. Xu, and J. Yu, "SportsCap: Monocular 3D human motion capture and fine-grained understanding in challenging sports videos," *Int. J. Comput. Vis.*, vol. 129, no. 10, pp. 2846–2864, 2021.

[9] P. Parmar and B. T. Morris, "What and how well you performed? A multitask learning approach to action quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 304–313.

[10] P. Parmar and B. T. Morris, "Learning to score Olympic events," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 20–28.

[11] P. Parmar and B. Morris, "Action quality assessment across multiple actions," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1468–1476.

[12] Q. Lei, H. Zhang, and J. Du, "Temporal attention learning for action quality assessment in sports video," *Signal, Image Video Process.*, vol. 15, pp. 1575–1583, Oct. 2021.

[13] E. E. Cust, A. J. Sweeting, K. Ball, and S. Robertson, "Machine and deep learning for sport-specific movement recognition: A systematic review of model development and performance," *J. Sports Sci.*, vol. 37, no. 5, pp. 568–600, 2019.

[14] A. Hall, B. Victor, Z. He, M. Langer, M. Elipot, A. Nibali, and S. Morgan, "The detection, tracking, and temporal action localisation of swimmers for automated analysis," *Neural Comput. Appl.*, vol. 33, pp. 7205–7223, Nov. 2020.

[15] C. Walker, P. Sinclair, K. Graham, and S. Cobley, "The validation and application of inertial measurement units to springboard diving," *Sports Biomech.*, vol. 16, no. 4, pp. 485–500, Oct. 2017.

[16] M. Brodie, A. Walmsley, and W. Page, "Fusion motion capture: A prototype system using inertial measurement units and GPS for the biomechanical analysis of ski racing," *Sports Technol.*, vol. 1, no. 1, pp. 17–28, Jan. 2008.

[17] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *Proc. BMVC*, vol. 2, 2010, p. 5.

[18] B. Fricke and T. Kothe, *Wasserspringen: Einblicke in Die Sporttechnik Und Ihre Vermittlung*, vol. 12. Aachen, Germany: Meyer & Meyer Verlag, 2009.

[19] C. Zhang, A. Gupta, and A. Zisserman, "Temporal query networks for fine-grained video understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4486–4496.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[23] M. Jain, J. Van Gemert, H. Jegou, P. Bouthemy, and C. G. M. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 740–747.

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[26] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster R-CNN architecture for temporal action localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1130–1139.

[27] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5822–5831.

[28] K. Soomro, H. Idrees, and M. Shah, "Action localization in videos through context walk," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3280–3288.

[29] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–9.

[30] M. Rashid, H. Kjellstrom, and Y. J. Lee, "Action graphs: Weakly-supervised action localization with graph convolution networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 615–624.

[31] P. Ghosh, Y. Yao, L. S. Davis, and A. Divakaran, "Stacked spatio-temporal graph convolutional networks for action segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 576–585.

[32] F. Carrara, P. Elias, J. Sedmidubsky, and P. Zezula, "LSTM-based real-time action detection and prediction in human motion streams," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27309–27331, Oct. 2019.

[33] T. Lin, X. Zhao, and Z. Fan, "Temporal action localization with two-stream segment-based RNN," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3400–3404.

[34] J. Mihir, L. Zhenyang, E. Gavves, and C. G. M. Snoek, "Action localization in sequential data with attention proposals from a recurrent network," U.S. Patent 15 250 755, Sep. 14, 2017.

[35] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 375–389, 2018.

[36] R. De Geest and T. Tuytelaars, "Modeling temporal structure with LSTM for online action detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1549–1557.

[37] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[38] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 156–165.

[39] W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, "Artificial neural networks: Formal models and their applications—ICANN," in *Proc. 15th Int. Conf. Artif. Neural Netw.*, Warsaw, Poland, Sep. 2005.

[40] R. Dai, S. Das, L. Minciullo, L. Garattoni, G. Francesca, and F. Bremond, "PDAN: Pyramid dilated attention network for action detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2970–2979.

[41] Y. A. Farha and J. Gall, "MS-TCN: Multi-stage temporal convolutional network for action segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3575–3584.

[42] K. Vats, M. Fani, P. Walters, D. A. Clausi, and J. Zelek, "Event detection in coarsely annotated sports videos via parallel multi receptive field 1D convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 882–883.

[43] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[44] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Amsterdam, The Netherlands: Springer, Jun. 2021, pp. 483–499.

[45] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "HigherHRNet: Scale-aware representation learning for bottom-up human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5386–5395.

[46] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, and W. Liu, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Apr. 2021.

[47] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[48] B. Artacho and A. Savakis, "UniPose: Unified human pose estimation in single images and videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7035–7044.

[49] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," 2022, *arXiv:2204.12484*.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[51] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, Z. Wang, and A. V. Den Hengel, "Poseur: Direct human pose regression with transformers," in *Proc. Eur. Conf. Comput. Vis.* Tel Aviv, Israel: Springer, Oct. 2022, pp. 72–88.

[52] K. Ludwig, S. Scherer, M. Einfalt, and R. Lienhart, "Self-supervised learning for human pose estimation in sports," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–6.

[53] T. Kitamura, H. Teshima, D. Thomas, and H. Kawasaki, "Refining OpenPose with a new sports dataset for robust 2D pose estimation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2022, pp. 672–681.

[54] J. Hong, M. Fisher, M. Gharbi, and K. Fatahalian, "Video pose distillation for few-shot, fine-grained sports action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9234–9243.

[55] N. Giulietti, A. Caputo, P. Chiariotti, and P. Castellini, "SwimmerNET: Underwater 2D swimmer pose estimation exploiting fully convolutional neural networks," *Sensors*, vol. 23, no. 4, p. 2364, Feb. 2023.

[56] J. Xu, Y. Rao, X. Yu, G. Chen, J. Zhou, and J. Lu, "Finediving: A fine-grained dataset for procedure-aware action quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 2949–2958.

[57] S. Honari, V. Constantin, H. Rhodin, M. Salzmann, and P. Fua, "Temporal representation learning on monocular videos for 3D human pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6415–6427, May 2023.

[58] V. M. Zatsiorsky and V. M. Zaciorskij, *Kinetics of Human Motion*. USA: Human Kinetics, 2002.

[59] V. Zatsiorsky, "Methods of determining mass-inertial characteristics of human body segments," *Contemporasy Problems Biomechnics*. Moscow, Russia: Mir Publishers, 1990.

[60] P. de Leva, "Adjustments to Zatsiorsky–Seluyanov's segment inertia parameters," *J. Biomech.*, vol. 29, no. 9, pp. 1223–1230, 1996.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[62] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: Video inference for human body pose and shape estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5253–5263.

PRAMOD MURTHY received the B.E. degree in computer science and engineering from S.R.T.M. University, India, in 2005, and the M.S. degree in computer science from Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), Germany, in 2016, where he is currently pursuing the Ph.D. degree. From 2006 to 2011, he worked in different engineering roles on system software and natural language processing for Indic languages. Since 2016, he has been a Research Assistant with the Department of Augmented Vision, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern. His current research interests include modeling human motion dynamics from monocular images and uncertainty estimation using Bayesian deep learning.

BERTRAM TAETZ received the B.Sc. and M.Sc. degrees in applied mathematics with physics as a minor subject and the Ph.D. degree in applied mathematics from Ruhr University Bochum, Germany, in 2009 and 2012, respectively. During his Ph.D. study, he focused on numerical methods for dynamical systems. Since 2013, he has also been a Senior Researcher with the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany. He was with the Department of Computer Science, University of Kaiserslautern (now Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau), from 2015 to 2022. Since 2022, he has been a Professor with the Department of IT and Engineering, IU International University of Applied Sciences, Germany. His current research interests include machine learning, sensor fusion, and computer vision methods related to human motion capturing and analysis.

ARPIT LEKHRA received the B.Tech. degree in information technology from the Indian Institute of Information Technology Allahabad (IIITA), India, in 2014. He is currently pursuing the master's degree in computer science with Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), Germany. From 2014 to 2018, he was a Software Developer and an Analyst role in the industry. His current research interest includes human pose estimation from monocular images.

DIDIER STRICKER led the Department of Virtual and Augmented Reality, Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, from 2002 to 2008. He is currently a Professor with the Department of Computer Science, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), Germany. He is also the Scientific Director of the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, where he leads the Augmented Vision Research Group. His research interests include 3D computer vision, autonomous driving, wearable health, augmented reality applications, and deep learning. He received the Innovation Prize from the German Society of Computer Science, in 2006. He serves as a reviewer for noteworthy journals in the area of VR/AR and computer vision.