Towards Competitive Diving Video Understanding

Xin Chen[1] Anqi Pang[1] Yang Wei[2]
Peihao Wang[1] Ge Zhang[1] Xuming He[1] Jingyi Yu[1]
[1]ShanghaiTech University [2]University of Delaware

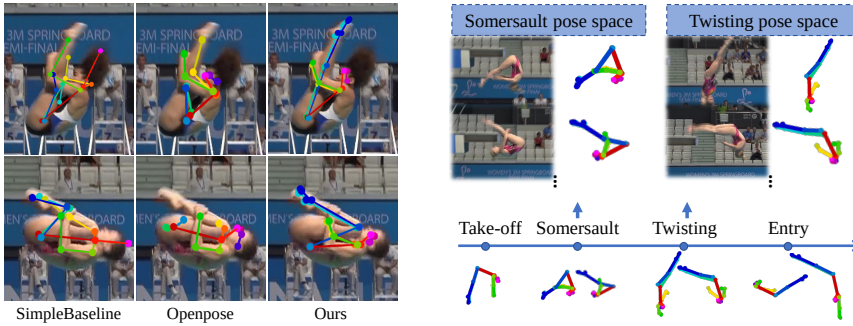
Abstract. Competitive diving videos impose significant challenges to existing video understanding techniques: moves such as twists and twists exhibit complex topology and occlusion patterns, and moves such as half and full somersaults appear similar but very different in scoring. We present a novel solution called DiveNet for understanding diving videos. For pose estimation, DiveNet explores the sequential constraint of sub-moves in a dive and constructs a reliable pose embedding network to infer pose parameters and skeletons in individual frames. For action analysis, we treat semantic implications of the dive number as attributes of the move and adopt a Spatial-Temporal Graph Convolutional Network (ST-GCN) with the pose parameter, joint and bone streams adapted structure for attribute prediction from the pose sequences and subsequent action labeling. We also build a new diving video dataset of 110K frames with the ground truth dive number, score, and skeleton. Comprehensive experiments on the new and existing datasets demonstrate DiveNet outperforms the state-of-the-art in both accuracy and robustness.

Keywords: Semantic action decomposition, Pose embedding, Sport video understanding

1 Introduction

The problem of image/video understanding for competitive sports such as diving and skating has attracted much attention in recent years. Applications are numerous, ranging from correcting athletes' movements to improve their performance to digitally producing 3D avatars for video games and feature films. Compared with video understanding under the general setting (e.g., walking), moves in competitive sports are fast and complex but are also more restrictive by following specific rules. In competitive diving, for example, divers perform challenging moves such as somersaults and twists, all exhibiting occlusion and topology patterns that are difficult to analyze using standard pose/action analyzers.

In this paper, we present a novel solution that we call DiveNet for understanding competitive diving videos by automatically identifying the dive number, assessing the performance, 2D pose/action recovery, etc. The problem resembles traditional pose estimation and action analysis but imposes unprecedented challenges. For pose estimation, despite tremendous advances in recent years, most existing techniques have still been focused on modeling relatively simple poses with few occlusions. In contrast, most diving moves exhibit heavy occlusions. For example, a somersault or a flip requires the diver to rotate 360 degrees around a horizontal axis with the feet passing over the head.



(a) Sample testing of diving (b) Semantic diving pose space.

Fig. 1. (a) Pose estimation on a sample frame in dive video using the top-down method, SimpleBaseline[45]) trained on MPII[1] dataset, the bottom-up method, OpenPose[5]on COCO[20] dataset, and DiveNet on our dataset. Here, We intended to show that without proper datasets and reliably network designs, state-of-the-art techniques can easily fail on competitive sports videos. All experiments in Sec. 4 still use the same training data. (b) The pose space of each diving sub-moves. We use semantic action label, which we call sub-move, to constrain the pose space, like somersault and twisting in our case.

Hence a large portion of the body is occluded from either frontal or side views, and the move can be performed forwards, backwards, or sideways for which the athletes themselves require decades of (physical) training to master. Action analysis is even more challenging: half somersault and full somersault are difficult to distinguish even if individual poses at frames are recovered.

DiveNet tackles the problem from two fronts. On pose estimation, we observe the move in competitive diving follows four stages or sub-moves: take-off, twist, somersault, and entry, where each stage contains complex yet restricted pose sequences. This indicates poses within each sub-move lies in a low-dimensional pose space. We hence construct a Pose Embedding Module composed of pose prediction branches for each sub-move. Each frame is fed into its respective branch to obtain its pose parameter, joints, and bones. For action recognition, instead of a training a black-box solution to map the pose sequence to dive number, we treat semantic implications of the dive number as attributes of the complete action and construct a 3-stream Spatial-Temporal Graph Convolutional Network (ST-GCN)[17] that takes the parameters, joints, and bones as input and conduct multi-task action attribution prediction. Finally, we adopt a Semantic Attributes Mapping Block to infer the action labels (dive number) from the attributes.

For training and evaluation, we first build a new diving video dataset composed of 640 video clips with a total of 110K frames. Each clip has the ground truth dive number and final score and is pre-segmented into sub-moves. For each frame, we manually label the bounding box of the athletes and their skeleton, including invisible joints. We validate the effectiveness and accuracy of our approach for pose estimation and action analysis on the new dataset as well as action analysis alone on the AQA dataset[27] which contains difficulty labels and scores but not poses. We show our approach out-

performs the state-of-the-art solutions including the latest C3D-LSTM [28] and R2+1D [42] in both accuracy and robustness. In addition, the poses produced from DiveNet can be used to drive a 3D avatar for performing the same dive as in the videos. Using Simplify-X[29], we generate reliable 3D diving suitable for virtual and augmented reality experience.

2 Related Work

Our work is closely related to pose estimation and action parsing, although diving videos exhibit new challenging issues such as complex occlusion patterns, ambiguity in moves, ambiguity in actions, etc. Therefore we only discuss the most relevant ones.

Pose Estimation aims to recover the underlying kinematic structure of a person. Earlier methods either adopted geometric constraints [47] or treated the problem as classification/regression based on features[4, 34]. The availability of ultra-large scale, labeled, human image/video data such as COCO[20] and MPII[1] has enabled highly effective deep learning based approaches[5, 32, 33, 38–40]. Openpose[5] employs Part Affinity Fields (PAF) to support bottom-up estimation. [38] exploits multi-scale high-resolution networks to improve feature representation. Nearly all existing approaches have focused on regular movements and actions rather than the ones in professional sports. A few recent approaches aim to tackle special actions. [23] proposes a semantic-based, multi-task learning framework and [3] tailors a predictor specific to certain actions. [46] uses a hierarchical structure to decompose an action into sub-poses and further divides them into parts. Their approach does not consider rich semantic information embedded in sports and the ordering constraints between sub-poses. In contrast, our approach explicitly uses the underlying semantic and ordering constraints.

Action Parsing can be categorized into short vs. long dynamics, depending on the length of the motion patterns. For short term dynamics, [15] uses 2D CNNs to learn deep appearance features and conduct frame-level classification. IDT[10] extends the technique with shallow motion features and [13] uses 3D CNNs such as C3D[41] to capture spatial-temporal patterns of consecutive frames within the sequence. For long term dynamics, TRN[50] exploits temporal dependencies across video frames over multiple hierarchies. TRN[50] proposes a multi-stream architecture to extract even richer temporal features. LTC[43] treats the temporal resolutions as a substitute to temporal windows whereas [13] conducts long-range action recognition. We observe that diving is a mixture of long and short dynamics: sub-actions such as twisting or somersaults map to short dynamics whereas the complete dive, with a corresponding dive number, map to long dynamics. We hence combine the advantages of short and long dynamics techniques.

To tackle diving videos [19], [14] employs attentive spatial-temporal representations and performs fine-grained motion recognition. [31] combines dense motion trajectories and pose estimation to improve recognition accuracy. Accuracy in recognition depends heavily on the viewpoint of the diver. [7] partitions the video into segments to improve action analysis but falls short of pose estimation under complex occlusions. In a similar

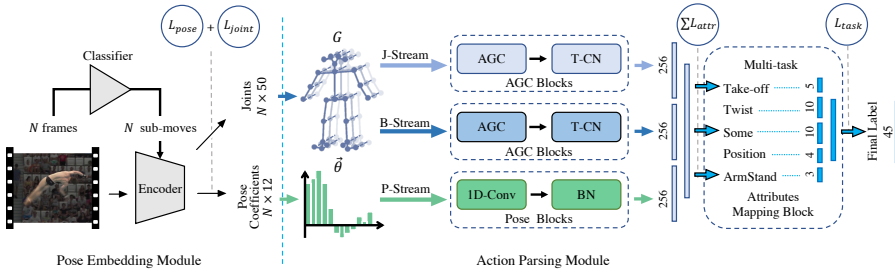


Fig. 2. Our DiveNet is composed of two main components: the Pose Embedding Module and the Action Parsing Module.

vein, [9] applies sub-action recognition for hockey. Our approach resorts to the recent Graph Convolutional Network (GCN) [8, 11] to handle spatial-temporal representations via ST-GCN[17, 18, 35].

By far, very few approaches have sought to exploit the rich semantic information: each dive corresponds to a four-digit number whose semantic constraints can be used to segment the video sequence as well as to restrict the pose/action space. In addition, none of the existing diving datasets contains per-frame labeled human pose as we have in our data. Our solution also provides effective Action Assessment for evaluating the quality of a move. [30] extracts pose features on the complete move and applies linear vector regression of assessing the move. Most recently, [26, 27] presents a new MTL-AQA dataset that exploits multi-task networks along with a caption generation model to simultaneously assess the move and produce a caption. [25] focuses on the anomaly in actions via joint relation graphs.

3 Pose and Action Analysis

Fig. 2 shows the architecture of our DiveNet. It takes a dive sequence as input and feeds into an end-to-end multi-task network. DiveNet consists of two modules for per-frame pose estimation and per sub-action labeling. It also conducts action assessment, and the pose estimation results can be directly used to drive a 3D avatar to conduct the same dive move. For pose estimation, we assume the clip corresponds to the complete dive move and first partition it into four segments using [12] that correspond to take-off, twist, somersault, and entry. For pose estimation, we construct their corresponding pose embedding branches for respective segments where each frame is fed to its respective branch to obtain its pose parameters, joints, and bones. For action labeling, we construct a 3-stream ST-GCN that takes the parameters, joints, and bones as input and conduct multi-task action attribution prediction. Our ST-GCN contains an attributes mapping block that assembles action attributes into the final label.

3.1 Pose Embedding

Complex occlusion and topology patterns impose significant challenges to existing pose estimators such as OpenPose[5] and SimpleBaseline[45]. Fig. 1 shows some typical results. This is partially due to lack of training data as the current mainstream human pose datasets (e.g., COCO[20] and MPII-Pose[1]) do not cover the pose variants in sports. More importantly, those heatmap-based approaches ignore the semantic and structural constraints specific to diving.

We present a novel Pose Embedding Module (PEM) that takes into account the semantic meaning of the move. We recognize that the complete dive move in professional diving always follows four stages: take-off, twist, somersault, and entry that we call sub-moves. We thus set out to construct a parametric pose model for each sub-move as the pose (sequence) within the sub-move exhibits high resemblance across athletes, e.g., they straighten their bodies in twisting while curling up in somersault.

PEM aims to build a parametric pose space with specific constraints for each sub-move. We represent the human pose as a set of 2D joint positions $\mathbf{J}$. Each sub-move m corresponds to a mapping:

$$\mathbf{J} = \mathcal{M}(\boldsymbol{\theta}; m) \tag{1}$$

where $\mathcal{M}(\boldsymbol{\theta}; m) : \mathbb{R}^K \mapsto \mathbb{R}^{2N}$, N denotes joint number and K denotes the dimension of parameters $\boldsymbol{\theta}$. We adopt the Principal Component Analysis (PCA) to model the pose space of each sub-move. For each m, we collect N labeled images to form a pose set $\mathcal{J}^m$. We conduct PCA on the pose set and generate a set of pose bases $\mathbf{B}^m = \{\mathbf{b}_k^m\}_{k=0}^K$ so that $\mathbf{J}$ under sub-move m can be represented as a weighted sum of the bases:

$$\mathbf{J} = \sum_{k=1}^{K} \theta_k \mathbf{b}_k^m + \mathbf{b}_0^m = \boldsymbol{\theta}^\intercal \mathbf{B}^m + \mathbf{b}_0^m \tag{2}$$

where $\mathbf{b}_0^m$ is the mean pose and $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_K]^\intercal$ are the weights.

Fig. 3(b) visualizes the first three elements of the pose bases of two sub-moves, somersault, and entry, with semantic annotations for separate subsets. The approach can robustly handle all sub-moves even for traditionally challenging poses. The resulting pose representations also encode semantic meanings, important for subsequent action parsing Sec. 3.2.

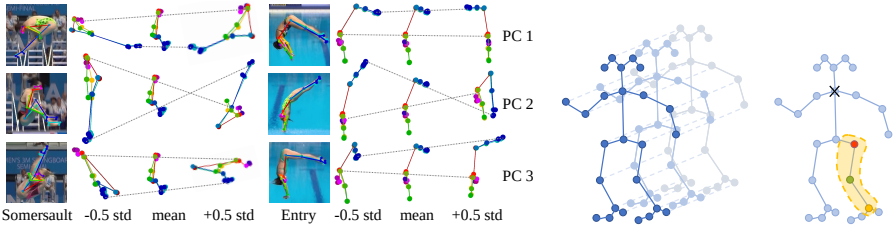
From the pose bases for all sub-moves, we construct the PEM that estimates the joints and bones (skeletons). PEM consists of a backbone convolutional encoder (e.g., ResNet-152) followed by 2 fully connected layers to regress the pose parameters $\boldsymbol{\theta}$ used to reconstruct the joint positions:

$$\boldsymbol{\theta}(\mathbf{x}) = \mathcal{F}_{conv}^m(\mathbf{x}; \mathbf{W}), \tag{3}$$

$$\mathbf{J}(\mathbf{x}) = \boldsymbol{\theta}(\mathbf{x})^\intercal \mathbf{B}^m + \mathbf{b}_0^m \tag{4}$$

where $\mathbf{x}$ denotes input image/frame and $\mathcal{F}_{conv}^m$ is the Pose Embedding Network for the sub-move class m.

Unlike prior approaches that target at general poses by implicitly encoding pose regularity into a complex network and hence cannot reliably handle motion blurs and



(a) PCA components of somersault/entry pose space (b) Our spatial–temporal graph

Fig. 3. (a) PCA pose spaces for two sub-moves: somersault and entry. For each sub-move, from left to right, we show the input frame and the first three principal components within 0.5 deviations from the mean. The lines connecting the corresponding elements within the component indicate the linear change according to the basis. (b) Left shows the spatial–temporal graph of the joints and bones. Right shows the spatial configuration partitioning strategy: we divide the node's one neighbor into three subsets: the root node (green dot), the centripetal subset (red dots), and the centrifugation subset (yellow dots), details in [17].

occlusions or easily enforce semantic constraints, PEM manages to exploit the structural constraints in diving poses with action semantics. Even though the paper focuses on diving, PEM can be applied to other types of sports such as ice skating.

3.2 Action Parsing

Next, we feed the estimated pose parameters and joint positions of all frames from PEM to the Action Parsing Module (APM) for analyzing the complete action, including inferring the dive number from the sequence and later assessing the performance. The brute-force approach would be to build a black-box network to map the pose sequence to dive number. Rather, we adopt a different approach that treats the dive number as attributes of the action. Recall the dive number encodes key semantic meaning of the dive move, e.g., the exact take-off type, the number of rounds of somersault and twist, etc. (see Fig. 5(a)), we call them semantic-attributes (SAs) and aim to learn how each frame contributes to respective attributes. Our APM explicitly recovers SAs via a two-stage architecture: in the first, we use a 3-Stream ST-GCN for SA predictions, and in the second, an attributes mapping block infers the dive number from the SAs.

Spatial-Temporal Feature Extraction. A number of previous approaches [44, 36, 49] exploit skeletons alone as inputs to the GCN. We observe in addition to skeletons (bones and joints), pose parameters obtained from PEM provide useful information on action parsing, as shown in Tab. 3. We therefore construct a 3-Stream convolutional module that takes joints (J-stream), bones (B-stream), and pose parameters (P-stream). For J- and B-Stream, we adopt the 2s-AGCN structure that can adaptively learn graph edge connections. Details on graph construction and partitioning are shown in Fig. 3(a). Specifically, we adopt the human joints and bones setting in OpenPose[5]. In the J-stream, the joints are mapped to graph nodes, and the bones maps to edges. In the B-Stream, the mappings are reversed. We feed 90 consecutive frames of skeletons into a 10-layer ST-GCN to extract two feature vectors. For the P-stream, we represent pose

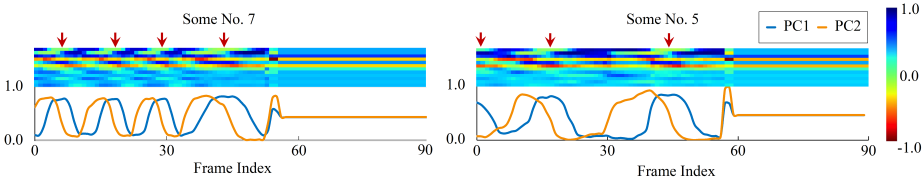


Fig. 4. Extracted feature maps of PCA pose parameters from the P-stream. Each feature map is of 90×12 (90 frames and 12 dimensional features). The first two principal components already reveal the number of rounds of half-somersault (left: $4 \times 2 - 1 = 7$ rounds; right: $3 \times 2 - 1 = 5$ rounds). Red arrows corresponds to the start position of a somersault (toe pointing to the ground) whereas the final half-somersault corresponds to entry to water.

parameters as a 1D vector and use layers of 1D convolution with residual blocks to generate features of 256 dimensions. We demonstrate the effectiveness of the proposed P-Stream. In Fig. 4, we visualize one of the feature maps of a specific sequence obtained from the P-Stream, at a resolution of 90×12 (90 frames and 12 dimensions in feature). We also plot the first two dimensions vs. frame index. We observe that they can be readily used to infer the somersault number. We finally concatenate all feature vectors generated by the J-, B- and P-Stream as inputs to the following attributes mapping block.

Semantic Attributes Mapping Block. The Semantic Attributes Mapping Block (SAMB) aims to learn the mapping between the extracted spatial-temporal features and dive numbers, i.e., to tell which dive type the video corresponds to. Instead of directly learning the mapping via a black-box solution, we sought to use Semantic Attributes (SAs) explicitly. Specifically, our goal is to partition the whole action sequence in terms of the SAs, or more precisely, how individual frames contribute to specific SAs. We use two fully connected layers to predict their contributions where the categories of all SAs are represented as vectors. Finally, we stack the resulting SAs and feed the results to another two fully connected layers to infer the dive number. Compared with black-box end-to-end approaches, our results show the use of intermediate SAs better supervise the training process, provide heuristic cues analogous to human labeling, and accelerate the training process.

3.3 Training Strategy

To train our network, we adopt a deeply-supervised strategy in which we design two losses for pose estimation module, PEM, and action parsing module, APM, respectively.

Loss for PEM. In our network, we use two types of representations, i.e., pose parameters and joints, to model skeletons. The pose parameters define the motion space of the skeleton, whereas the joint positions better describe the visibility between joints within an image. We therefore design the pose parameter loss $\mathcal{L}_{\boldsymbol{\theta}}$ as:

$$\mathcal{L}_{\boldsymbol{\theta}} = w \left\| \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right\|_1, \quad w = \frac{|\mathbf{V}|}{N} \tag{5}$$

where $\boldsymbol{\theta}$ is the ground truth pose parameters, w is the weight for joints visibility, $\hat{\boldsymbol{\theta}}$ is the predicted pose parameters and $\mathbf{V} \in \mathbb{R}^{2N}$ indicates the visibility of the ground truth joint (0 means invisible and 1 otherwise).

We design the joint position loss $\mathcal{L}_{\mathbf{J}}$ as:

$$\mathcal{L}_{\mathbf{J}} = \mathbf{V} \cdot (\mathbf{J} - \hat{\mathbf{J}}), \quad \hat{\mathbf{J}} = \mathcal{M}(\hat{\boldsymbol{\theta}}; a)$$

where $\mathbf{J}$ is the ground truth 2D joints, $\hat{\mathbf{J}}$ is the joints obtained through mapping $\hat{\boldsymbol{\theta}}$ to 2D joints. We combine the pose parameter and joint loss with a weight parameter λ_P as the final loss of the PEM:

$$\mathcal{L}_P = \mathcal{L}_\theta + \lambda_P \mathcal{L}_{\mathbf{J}} \tag{6}$$

Loss for APM. For the APM, we use the cross-entropy loss between the predicted and the ground truth attributes, which can be written as follows,

$$\mathcal{L}_{attr} = \sum_{c=1}^{Ns} \sum_{i=1}^{N_c} y_i^c \log\left(x_i^c\right) \tag{7}$$

where c indicates the attribute type, N_s denotes the number of attributes, and N_c is the number of classes within each attribute c. Here x_i^c denotes the prediction for the i-th label of attribute c whereas y_i^c is the ground-truth.

For the action labeling task, we also add the cross-entropy loss between the prediction and the ground truth action label as the task loss as below. We note that such a task loss depends on the target application and can be easily adjusted according to the final task.

$$\mathcal{L}_{task} = \sum_{j=1}^{N_f} y_j \log\left(x_j\right) \tag{8}$$

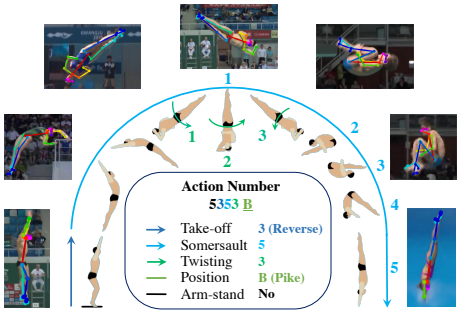
where N_f denotes the total number of all possible action labels (dive number), and x_j, y_j are the prediction and ground truth of the j-th final label.

The overall loss for the APM is a combination of the attribute loss and the task loss as following:

$$\mathcal{L}_A = \mathcal{L}_{attr} + \lambda_A \mathcal{L}_{task} \tag{9}$$

where λ_A is a weight to balance two loss terms.

Training. While our entire network can be trained in an end-to-end fashion, we exploit its modular architecture and develop a stage-wise strategy, which is more efficient in practice. Specifically, our training procedure is composed of three stages: 1) We train a PEM for each sub-move independently and fix its parameters, 2) We then train the action attribute prediction and label classification modules in the APM jointly, and 3) Finally we fine-tune the entire network using the combined losses of PEM and APM (i.e., $\mathcal{L}_P + \mathcal{L}_A$).



(a) Semantic attributes of DND.

Dataset	Videos	Images	Actions per video	Pose	Assess	Type
Ours	640	110K	Multi	✓	✓	Sport
Penn[48]	2K	160K	Single	✓	×	Sport
AQA[27]	1K	150K	Multi	×	✓	Sport
COCO[20]	-	330K	-	✓	×	General
MPII-Pose[1]	-	25K	-	✓	×	General

(b) Comparisons with existing datasets.

Fig. 5. (a) Our DiveNet Dataset (DND) and semantic attribute classes for a complete diving from take-off to entry water. (b) Comparison between DND and existing datasets regarding size, action labels of per video, pose annotation, action assessment, and pose type.

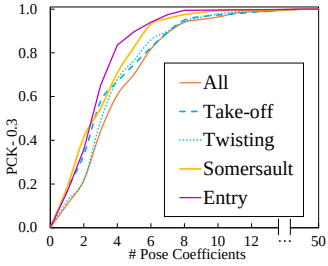
4 Experimental Results

To evaluate our DiveNet, we first construct a new dive video dataset called DiveNet Dataset (DND). Our DND contains rich poses and semantic action annotations. Fig. 5(b) compares DND with existing datasets. Existing human pose datasets, such as MPII[1] and COCO[20] focus on human pose in daily activity, whereas DND focuses on sports movements (see Fig. 1, 7). Other sports datasets either lack assessment scores such as PenAction[48] or human skeletons such as AQA[27]. We train and test the PEM on our own dataset. We then use the complete DiveNet for dive number classification and action assessment, on both DND and AQA dataset[27].

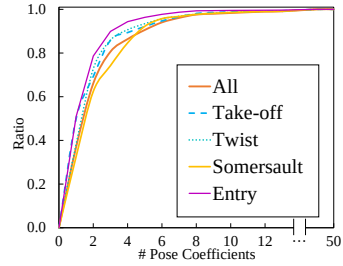
Our DND contains 640 videos (110K frames) with skeleton annotation, sub-move labels, and action assessment score. Specifically, each frame is annotated using five sub-move labels, including take-off, twisting, somersault, entry, and splash. There are 36,613 annotated skeleton (25 joints representation, same as Openpose[5]) with bounding box. In addition, we annotate the visibility of each joint as three types: visible, labeled but not visible, and not labeled (same as COCO [20]). We also include the SAs (see Fig. 5(a)) for each video, including the take-off type, twisting number, somersault number, arm-stand, and dive position. We include the difficulty score, the number of valid referees, the execution score, and the final assessment score. DND will be shared with the community as the first comprehensive dive video dataset.

4.1 Training and Evaluations

Training Details. We resize image patches that contain the human body at a resolution of 256×256 (using the ground truth bounding box in our DIVE dataset and detect the bounding box in AQA[27] using[21]). We re-sample the video to 90 frames each. For PEM training, we conduct data augmentation via random rotations ($-45°$ to $+45°$),



(a) PCK-0.3 vs. PCA dimension.



(b) The ratio vs. PCA dimension.

Fig. 6. Cumulative relative variance of our dataset explained as a function of the number of pose parameters. 'All' refers to use all training data rather than restricted to each semantic pose space.

Table 1. Accuracy of fine-grain classification on sub-moves for partitioning the video clips.

Method	Take	Twist	Some	Entry	Splash	Avg.
WS-DAN[12]	97.8	94.5	96.5	95.3	97.6	96.7

random scaling (0.7 to 1.3), and flipping horizontally. For APM training, we also augment the skeleton and pose parameters data for J-, B- and P-Streams, respectively. For J-Stream and B-Stream, we scale the joint positions via interpolation to simulate the far and near camera views. In addition, we randomly mirror the skeleton (to emulate viewing from both sides). For the P-Stream, we also scale the coefficients vector.

We use the Adam optimizer[16], train the PEM in the first 100 epochs and AMP in the following 50 epochs. We train the complete DiveNet in the last 10 epochs. The learning rates of the 0th, 70th, and 150th epoch are 10^{-3}, 10^{-4} and 10^{-5}, respectively. We train our DiveNet on 4 NVidia 2080Ti GPUs, and the process takes 10 hours for PEM, 5 hours for APM, and 2.5 hours for the whole DiveNet. Once trained, the network processes the $90 \times 256 \times 256$ video data at 0.5s for PEM, 0.05s for APM and 1.0s for the data fetching.

For fair comparisons, we re-train HRNet [38] and SimpleBaseline [45] using our DND dataset. For SCADC [27], C3D-LSTM [28], C3D-AVG [27], and R2+1D [42], we first pre-train the corresponding networks using the UCF101 dataset [37] and I3D [38] on the Kinetics dataset [38], replace their output or the regression layers with our proposed SAMB module, and fine-tune SAMB with our DND dataset.

PEM Evaluation. We use the classifier in [12] to first predict the sub-move label. We observe the technique can achieve high accuracy, and the predicted sub-move label helps the PEM for pose estimation. Tab. 1 shows our sub-move classification produces an average accuracy of 96.7%.

PCA analysis in Fig. 6 demonstrates that the poses of each sub-move lay in a low-dimensional space. As comparison with conducting PCA on the whole action data, we further reduce the space dimension by cooperating the semantic action labels. Fig. 6 shows that the pose space of each semantic exhibits lower-dimensionity. Tab. 2 shows that with semantic pose space, the accuracy increases by 1.8% on the PCK-0.3. We use

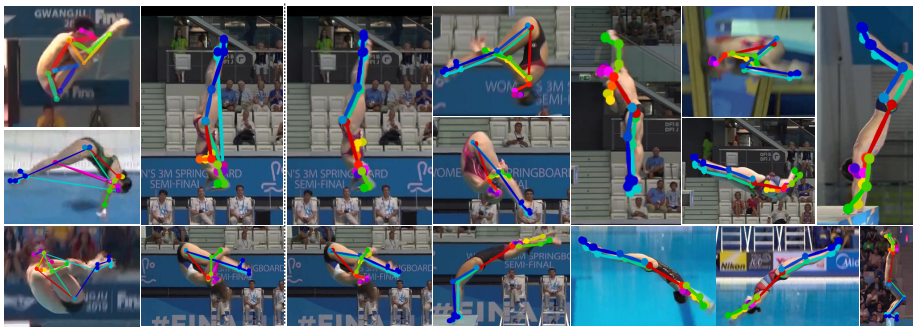


Fig. 7. Additional pose estimation results using SimpleBaseline, HRNet, and DiveNet. All methods are trained on the same DND dataset.

Table 2. Comparisons of different methods trained on the same DND dataset: the baseline (HRNet[38] and SimpleBaseline[45]) and ablation study (PCA w/o sub-move labeling, 50/101/152 ResNet as Backbone, and $\mathcal{L}_{\boldsymbol{\theta}}/\mathcal{L}_{\mathbf{J}}/\mathcal{L}_{\boldsymbol{\theta}}+\mathcal{L}_{\mathbf{J}}$ as loss for training).

Method	PoseSpace	Backbone	TakeOff	Twist	Some	Entry	PCK-0.3	PCK-0.5
Ours+$\mathcal{L}_{\mathbf{J}}$	-	ResNet50	69.1	60.7	76.5	71.3	70.9	86.5
Ours+$\mathcal{L}_{\boldsymbol{\theta}}$	Semantic	ResNet50	82.9	75.6	83.1	90.3	83.6	91.5
Ours+$\mathcal{L}_{\boldsymbol{\theta}}+\mathcal{L}_{\mathbf{J}}$	Single	ResNet50	81.0	80.1	82.7	89.8	83.0	92.1
Ours+$\mathcal{L}_{\boldsymbol{\theta}}+\mathcal{L}_{\mathbf{J}}$	Semantic	ResNet50	81.3	80.4	87.6	88.8	84.8	92.4
Ours+$\mathcal{L}_{\boldsymbol{\theta}}+\mathcal{L}_{\mathbf{J}}$	Semantic	ResNet101	**87.2**	82.2	88.6	90.5	87.5	94.5
Ours+$\mathcal{L}_{\boldsymbol{\theta}}+\mathcal{L}_{\mathbf{J}}$	Semantic	ResNet152	83.6	**84.6**	**91.5**	**94.0**	**88.5**	**96.0**
HRNet[38]	-	-	83.9	81.7	82.7	86.4	83.6	87.5
Simple[45]	-	ResNet152	86.3	68.5	86.7	90.4	84.2	88.9

the first 12 principle components as basis vectors to represent the pose space of each semantic action. A pose can be viewed as the weighted summation of the basis vectors. We also observe that the semantic sub-action recognition is highly accurate as in Tab. 1 and it benefits our semantic based pose space approach.

We evaluate the performance of the PEM in Tab. 2. We compare PEM with the state-of-the-art pose estimation work HRNet[38] and the SimpleBaseline[45]. For fair comparison, we re-train HRNet and SimpleBaseline these methods on the DND dataset. We consider the joints with distance errors less than 0.3 and 0.5 of the torso length as correct predictions (details in [2]) and report the percentage of correct keypoints (PCK-0.3 and PCK-0.5) as our metric. For the classifier, we test ResNet-50, -101, -152 as our backbone and find every 50 layers lead to above 2% change for PCK-0.3. Our approach has a similar performance as HRNet[38] and SimpleBaseline[45] for the take-off pose but can better handle other sub-moves with more complex occlusion patterns, as shown in Fig. 7. Our method leads by 4.3∼4.9% on PCK-0.3, but 7.1∼8.5% on PCK-0.5, this indicates our approach are reliable for generating basic location and the joints are more natural and stable.

Table 3. Action parsing evaluation using state-of-the-art approaches vs. our APM method with joint/joint+bone/joint+bone+pose(parameter) streams, on our DND and the AQA dataset.

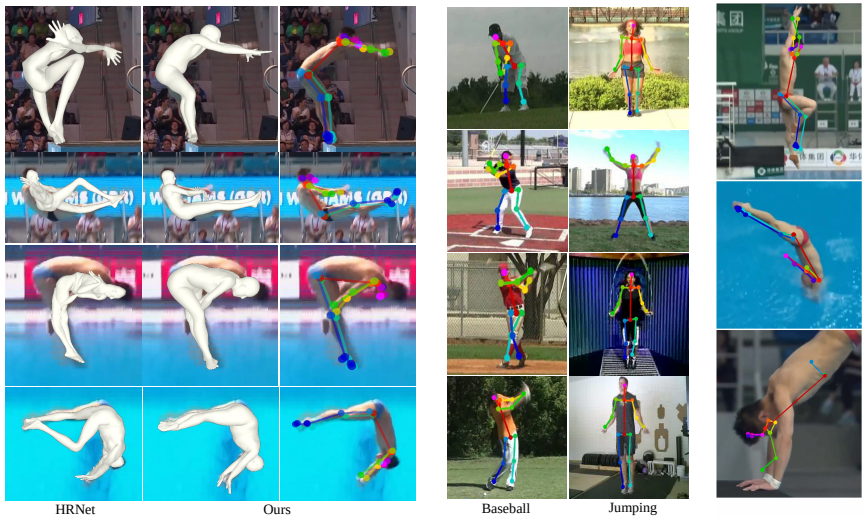
Dataset	Method	TakeOff	ArmStand.	Twist No.	Some No.	Position	Diving No.
DND	C3D-LSTM[28]	43.1	85.3	66.1	46.8	56.0	27.3
	R2+1D[42]	34.9	84.4	64.2	44.9	55.6	26.1
	I3D[6]	61.5	92.7	70.6	69.7	70.6	58.6
	Ours+J	85.3	97.5	78.0	82.9	75.6	65.0
	Ours+J+B	84.1	98.6	76.8	**90.2**	84.7	67.1
	Ours+J+B+P Black-Box	-	-	-	-	-	78.0
	Ours+J+B+P SAMB	**96.4**	**99.8**	**89.5**	86.5	**92.6**	**82.2**
AQA[27]	Nibali[24]	74.8	98.3	78.7	77.3	79.9	-
	MSCADC[27]	78.4	97.5	84.7	76.2	82.7	-
	C3D-AVG[27]	**96.3**	99.7	**97.5**	**96.9**	**93.2**	-
	Ours	95.1	**99.8**	90.7	88.9	92.5	-

APM Evaluation. Next, we evaluate the APM module using the outputs of PEM. The 3-stream structure along with SAs of the APM module enables faster convergence. The 2s-AGCN structure (J- and B-Stream only) takes 70 epochs to converge, whereas the 3-stream structure converges after only 50 epochs. With SAMB, we further accelerate convergence to 10 epochs. The result shows the use of P-Stream (Tab. 3 row 6) significantly improves the accuracy vs. baseline (Tab. 3 row 4, 5), even though the accuracy on the somersault attribute drops. This is expected as we did not train specifically for somersault alone but all attributes at once.

We further conduct experiments to illustrate the benefits of SAMB over black-box. Specifically, we keep the 3-stream backbone but replace SAMB with black-box without using attribute loss $\mathcal{L}_{attr}$. The network with SAMB converges after 10 epochs with the final accuracy of 82.2% whereas the one with black-box converges after over 30 epoch with the final accuracy of 78.0%.

Finally, we compare the overall action parsing performance with methods including MSCADC[27], C3D-LSTM[28], C3D-AVG[27],I3D[6] and R2+1D[42]. We compare DiveNet with C3D-LSTM and R2+1D on only DND for dive number prediction. We did not test on MSCADC, C3D-AVG, and Nibaili[24] since they do not have the ground truth Dive Number. DiveNet achieves the highest accuracy. On the AQA dataset, we also compare DiveNet with Nibaili[24]. DND achieves similar performance as C3D-AVG but outperforms MSCADC and Nabaili. It is important to note C3D-AVG is not yet open source, but their performance was directly adopted from their paper.

Failure case. By tailoring our network for specific subset of moves, our approach may generate erroneous estimations on degraded images. Fig. 8(c) shows some examples such as when the image does not fall into any of the pre-defined pose categories, some body parts are invisible (e.g, head entry into the water), or incomplete images. We note that baseline algorithms in those cases may perform better as they enforce the final estimation to obey more general pose settings.



(a) Driving a 3D avatar. (b) Estimating other sports (c) Failure cases

Fig. 8. (a) The estimated pose can be used to drive a 3D avatar. We use the same SMPL model inferred from the rest pose as the 3D avatar for both HRNet and our PEM. (b) Extensions of our approach to other sports by training PEM on PennAction[48]. (c) Failure cases caused by missing training data and occlusions/image clipping.

Table 4. Action assessment results using the state-of-the-art vs. our DiveNet on DND and AQA.

Metric	Method	DND Dataset	AQA Dataset
Sp. Cor.	C3D-LSTM[28]	53.7	84.9
	R2+1D[42]	55.6	**89.6**
	Ours	**61.7**	86.2

4.2 Other Applications

Based on the pose and action analysis results, we can further conduct action assessment and drive a SMPL[22] model, e.g., for virtual and augmented reality applications.

Action Assessment. Action assessment in diving depends on two independent scores, the difficulty score determined by the dive number, and the execution score representing the quality of the action and the splash. We reuse the extracted spatial-temporal features for action quality assessment. To evaluate the splash, we use a pre-trained C3D block[41] on the UCF101 dataset[37] to extract a 4,096 dimensional feature. We concatenate this feature to our spatial-temporal feature for the final execution score assessment. Instead of treating it as a regression problem, we discretize the score range 0-100 to 49 labels evenly. We use the cross entropy loss Equ. 7 and conduct the same training strategy with the SA learning. We train our PEM module on our dataset for pose, while training and testing APM for execution score and final score on our dataset and AQA dataset respectively, Tab. 4 shows the result of the testing results. The reported execution score reaches 61.7% spearman's rank correlation (Sp. Cor.)[30].

Driving a 3D Avatar. Our pose estimation method can well handle the complex pose and motion blur problem in the diving action (see Fig. 7). We use our PEM output skeleton to drive a 3D human body and animate the diving action (see Fig. 8(a)). Specifically, we optimize the SMPL [29] using our inferred 2D pose sequences. Our animated 3D human faithfully resembles the diving move with smooth and natural transitions whereas results based on prior art exhibit jitters and unnatural moves. .

Other sports. Our technique can be potentially extended to process other types beyond diving, e.g., by training on the PennAction datasets. Fig. 8(b) shows sample results using SAMB for pose estimation on figure skating and gymnastics. However, unlike diving, we have not yet constructed the corresponding datasets with all sub-action labels and scores and therefore we cannot yet perform additional tasks such as auto scoring and action parsing.

5 Conclusions and Discussion

We have presented a novel method named DiveNet for understanding challenging competitive dive videos. Different from previous "one-network-for-all" approaches, DiveNet exploits unique movement ordering in diving, and we design tailored pose estimation networks for take-off, twist, somersault, and entry. For action analysis, instead of directly mapping the pose sequence to its corresponding dive number, DiveNet treats the dive number as attributes to the action to resolve ambiguity, such as single vs. double twists and half vs. single somersaults. Specifically, it employs ST-GCN to predict the attribute of individual frames and then infer the action as well as performance scores from all attributes. In addition to techniques, we have presented a new manually annotated video dataset, including dive number, score, and skeletons that will be public to the community.

Our approach, in essence, exploits the semantic action analysis, for human pose estimation. This is different from HRNet[39] or SimpleBaseline[45] that separately predict individual joints. Consequently, our network, once trained, only tackles specific sports rather than general movements as in prior art. Our current setup assumes a single video stream (i.e., the side view) as input. We observe that side views are more common in diving videos, for reasons: they embed richer information of the pose and incur less occlusions (e.g., in the front view, the torso often completely occludes legs). In fact, professional sport judges confirmed they use mostly the side views for scoring. Our dataset hence contains more side views than front views. In sports, it is common practice to show two or more video streams. In the future, we plan to combine multiple streams for a 3D pose/shape task. In addition, the general human activities can always be separate into small sub-moves. Hence we plan to extend our work to general-purpose pose estimation through human action decomposition, including to expand our DND dataset as a more general dataset. Finally, our solution has not taken into account the shape of the human body. Techniques in human shape modeling such as SMPL[22] may be potentially used to infer both skeleton and shape of the diver, to produce more realistic 3D animation of the dive.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2014)
2. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. pp. 3686–3693 (2014)
3. Bertasius, G., Feichtenhofer, C., Tran, D., Shi, J., Torresani, L.: Learning discriminative motion features through detection. arXiv preprint arXiv:1812.04172 (2018)
4. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: European Conference on Computer Vision. pp. 717–732. Springer (2016)
5. Cao, Z., Martinez, G.H., Simon, T., Wei, S., Sheikh, Y.A.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2019). https://doi.org/10.1109/TPAMI.2019.2929257
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
7. Choutas, V., Weinzaepfel, P., Revaud, J., Schmid, C.: Potion: Pose motion representation for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7024–7033 (2018)
8. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems 29. pp. 3844–3852 (2016)
9. Fani, M., Neher, H., Clausi, D.A., Wong, A., Zelek, J.: Hockey action recognition via integrated stacked hourglass network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 29–37 (2017)
10. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1933–1941 (2016)
11. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
12. Hu, T., Qi, H.: See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. arXiv preprint arXiv:1901.09891 (2019)
13. Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 254–263 (2019)
14. Kanojia, G., Kumawat, S., Raman, S.: Attentive spatio-temporal representation learning for diving classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
15. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference for Learning Representations (2015)
17. Li, C., Cui, Z., Zheng, W., Xu, C., Yang, J.: Spatio-temporal graph convolution for skeleton based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
18. Li, R., Wang, S., Zhu, F., Huang, J.: Adaptive graph convolutional neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)

19. Li, Y., Li, Y., Vasconcelos, N.: Resound: Towards action recognition without representation bias. In: Proceedings of the European Conference on Computer Vision. pp. 513–528 (2018)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the European Conference on Computer Vision. pp. 740–755. Springer (2014)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
22. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
23. Luvizon, D.C., Picard, D., Tabia, H.: 2d/3d pose estimation and action recognition using multitask deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5137–5146 (2018)
24. Nibali, A., He, Z., Morgan, S., Greenwood, D.: Extraction and classification of diving clips from continuous video footage. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 38–48 (2017)
25. Pan, J.H., Gao, J., Zheng, W.S.: Action assessment by joint relation graphs. In: Proceedings of the IEEE International Conference on Computer Vision (October 2019)
26. Parmar, P., Morris, B.: Action quality assessment across multiple actions. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision. pp. 1468–1476. IEEE (2019)
27. Parmar, P., Morris, B.T.: What and how well you performed? a multitask learning approach to action quality assessment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2019)
28. Parmar, P., Tran Morris, B.: Learning to score olympic events. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 20–28 (2017)
29. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)
30. Pirsiavash, H., Vondrick, C., Torralba, A.: Assessing the quality of actions. In: European Conference on Computer Vision. pp. 556–571. Springer (2014)
31. Pishchulin, L., Andriluka, M., Schiele, B.: Fine-grained activity recognition with holistic and pose based features. In: Proceedings of the German Conference on Pattern Recognition. pp. 678–689. Springer (2014)
32. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4929–4937 (2016)
33. Raaj, Y., Idrees, H., Hidalgo, G., Sheikh, Y.: Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4620–4628 (2019)
34. Rogez, G., Weinzaepfel, P., Schmid, C.: Lcr-net: Localization-classification-regression for human pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3433–3441 (2017)
35. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12026–12035 (2019)
36. Si, C., Jing, Y., Wang, W., Wang, L., Tan, T.: Skeleton-based action recognition with spatial reasoning and temporal stack learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 103–118 (2018)

37. Soomro, K., Zamir, A.R., Shah, M.: A dataset of 101 human action classes from videos in the wild. Center for Research in Computer Vision (2012)
38. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
39. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Proceedings of the European Conference on Computer Vision. pp. 529–545 (2018)
40. Tang, Z., Peng, X., Geng, S., Wu, L., Zhang, S., Metaxas, D.: Quantized densely connected u-nets for efficient landmark localization. In: Proceedings of the European Conference on Computer Vision. pp. 339–354 (2018)
41. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4489–4497 (2015)
42. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)
43. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **40**(6), 1510–1517 (2017)
44. Wen, Y.H., Gao, L., Fu, H., Zhang, F.L., Xia, S.: Graph cnns with motif and variable temporal block for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8989–8996 (2019)
45. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision. pp. 466–481 (2018)
46. Xiaohan Nie, B., Xiong, C., Zhu, S.C.: Joint action recognition and pose estimation from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1293–1301 (2015)
47. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR 2011. pp. 1385–1392. IEEE (2011)
48. Zhang, W., Zhu, M., Derpanis, K.G.: From actemes to action: A strongly-supervised representation for detailed action understanding. In: Proceedings of the IEEE International Conference on Computer Vision (December 2013)
49. Zhang, X., Xu, C., Tian, X., Tao, D.: Graph edge convolutional neural networks for skeleton-based action recognition. IEEE Transactions on Neural Networks and Learning Systems (2019)
50. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of the European Conference on Computer Vision. pp. 803–818 (2018)