S3D: Stacking Segmental P3D for Action Quality Assessment

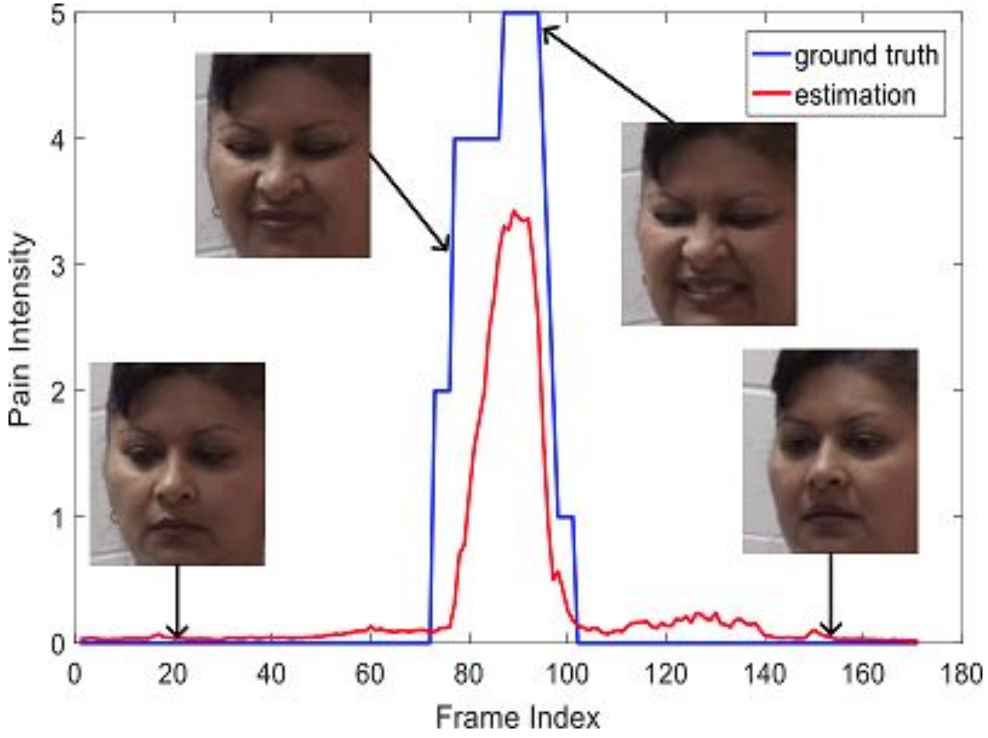
Presented by Trac D. Tran

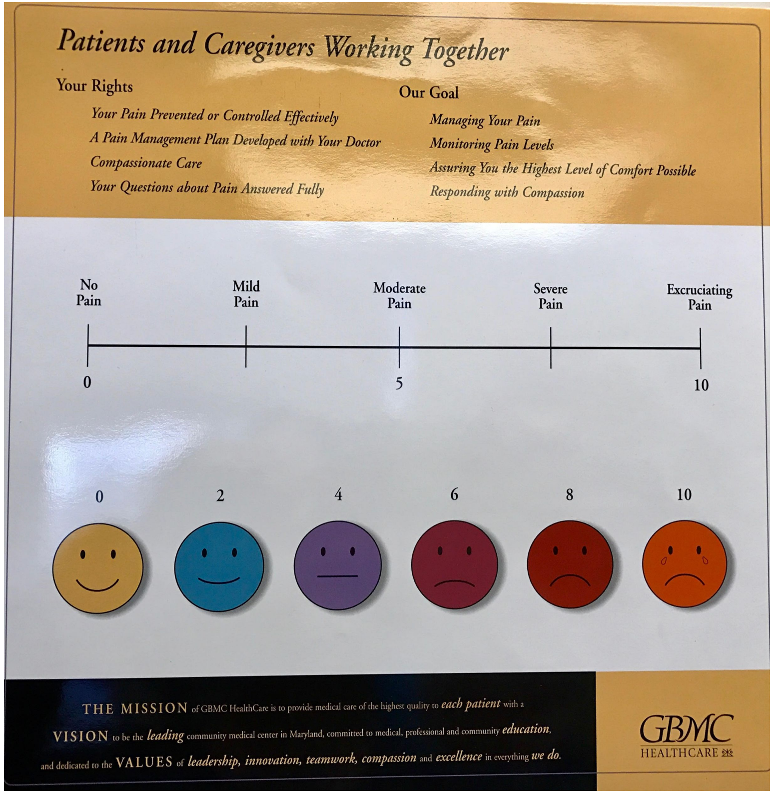
in representation of Xiang Xiang*, Ye Tian*, Austin Reiter, Grgeory D. Hager and Trac D. Tran

Johns Hopkins Univeristy, USA

(* Equal contribution)

Our Previous work: Action Classification->Intensity Regression





Feng Wang*, Xiang Xiang*, Chang Liu, Trac D Tran, Austin Reiter, Gregory D Hager, Harry Quon, Jian Cheng, Alan L Yuille. Regularizing Face Verification Nets For Pain Intensity Regression. In **IEEE ICIP, 2017**. (* Equal contribution)

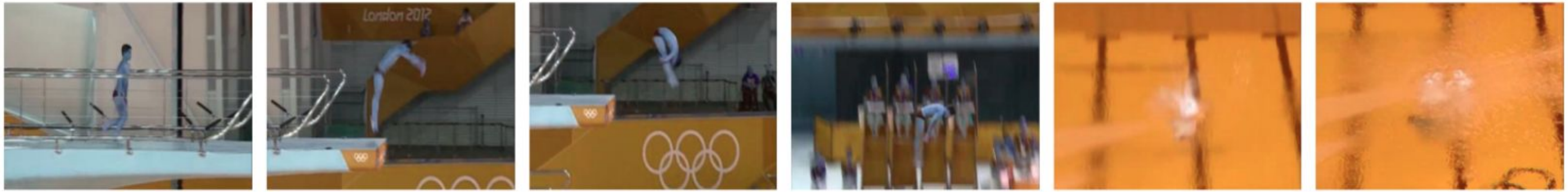
New Problem

Player: Chen
10 9.5 10 10 9.5 10 10
(3.6) 585.30

Player: Sanchez
7.5 8.0 8.0 8.0 8.0 7.5 8.5
(3.8) 532.70

Player: Boudia
6.5 6.0 6.0 6.5 4.5 6.0 6.5
(3.7) 525.25

Background: Video Representation Learning



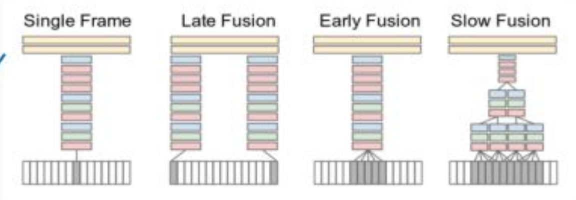
Video action assessment in our case is to predict a score s given a video $\mathbf{V}$ of one diving performance. As a supervised learning model, CNN is supposed to learn a mapping $f(\cdot)$ from $\mathbf{V}$ to s from training data such that $s = f(\mathbf{V})$. While it looks like that the action quality score is a function of the action video, essentially the video is a representation of the player's skill. As a result, there also exists $\mathbf{V} = g(s)$ where $g(\cdot)$ is a generative function: given a certain skill s, the generated action is recorded in the video $\mathbf{V}$. However, in this paper, we are trying to learn the underlying representation of the skill s from the video $\mathbf{V}$. Namely, if the mapping $f(\cdot)$ learned by CNN is good enough, it well characterizes the inverse video generation process as $s = f(\mathbf{V}) = f(g(s))$.

Existing Work: 3D CNN for Video Representation Learning

Video representation learning

2D Convolutional Neural Network

Large-scale Video Classification with Convolutional Neural Networks. [Karpathy, CVPR'14]



- Treat video as a bag of short, fixed-sized clips
- Extend the connectivity of the network in time dimension

Two-Stream Convolutional Networks for Action Recognition in Videos. [Simonyan, NIPS'14]

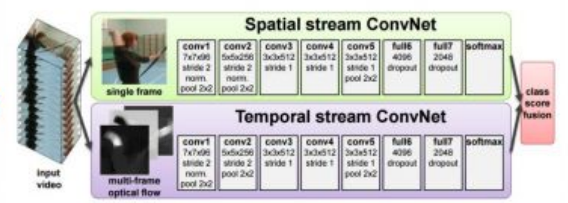
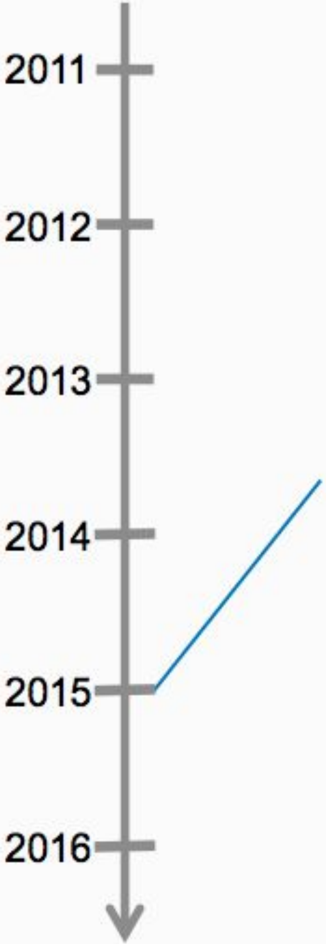


Figure 1: Two-stream architecture for video classification.

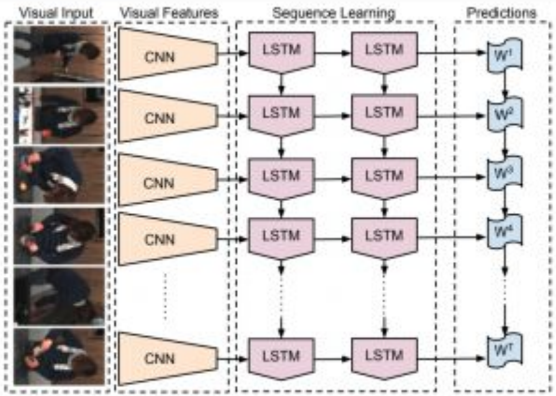
- Two-stream: frame + motion (stacked optical flow)
- 2D CNN for frame is pre-trained on ImageNet
- 2D CNN for motion is trained from scratch

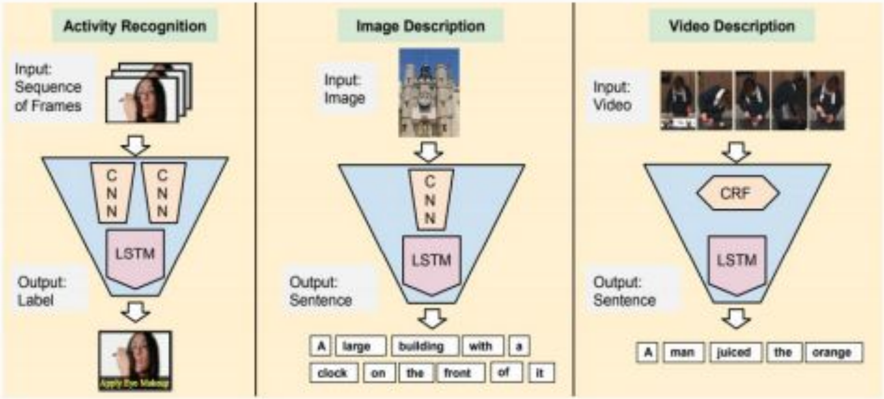
Video representation learning



2D CNN + LSTM (LRCN)

Long-term Recurrent Convolutional Networks for Visual Recognition and Description [Donahue, CVPR'15]

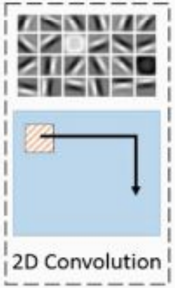




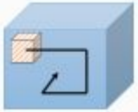
- Develop recurrent convolutional architecture
- Outputs of 2D CNN are fed into a stack of LSTM
- Applications on activity recognition and video description
- Neglecting low-level motion information

Video representation learning: from 2D CNN to 3D CNN

ResNet:
[MSRA, CVPR'16]



3D CNN:
[FAIR & NYU, ICCV'15]



Conv1a 64	Pool1	Conv2a 128	Pool2	Conv3a 256	Conv3b 256	Pool3	Conv4a 512	Conv4b 512	Pool4	Conv5a 512	Conv5b 512	Pool5	fc6 4096	fc7 4096	softmax	

Network comparison on Sports-1M

Network	Depth	Model Size	Video hit@1
ResNet	152	235 MB	64.6%
C3D	11	321 MB	61.1%
C3D	100+	~3 GB	--

- Training 3D CNN is very computationally **expensive**
- Difficult to train very **deep** 3D CNN
- **Fine-tuning** 2D CNN is better than 3D CNN

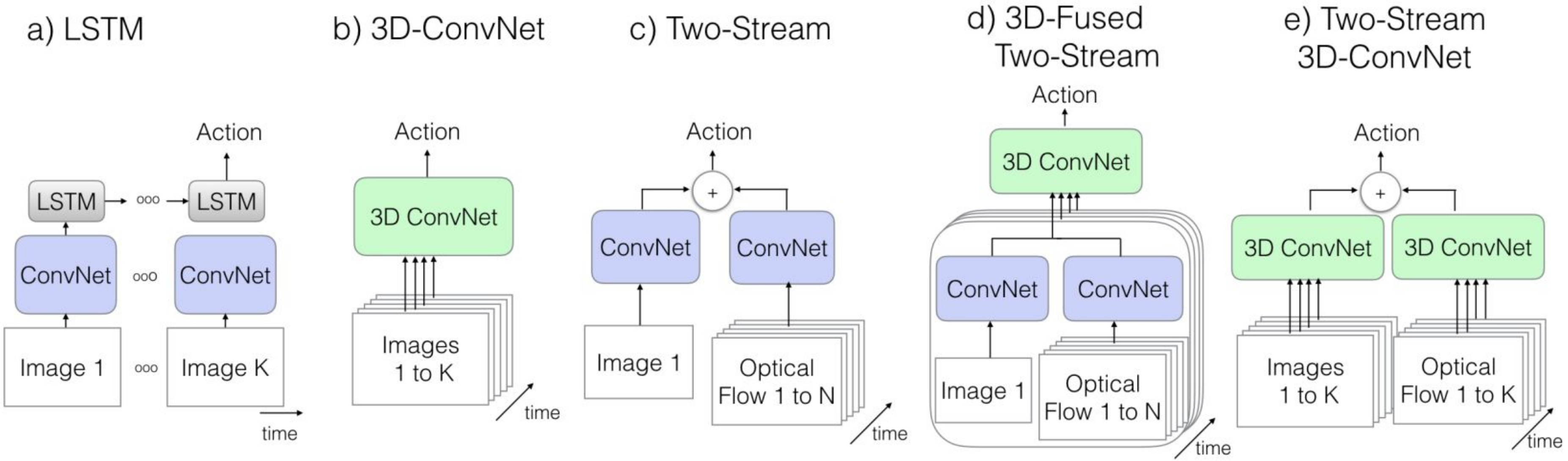
Intuition: 3D Convolution Factorization

Furthermore, for a convolution network, we have $\mathbf{I}(\mathbf{x}) = \sum_{i=1}^{r} \mathbf{u}_i(\mathbf{x}) * \mathbf{v}_i(\mathbf{x})$ where $*$ denotes the operation of convolution. The operation of u_i convolving with v_i can be written as the matrix convolution in the dimension of $n_{pixels} \times (n_{pixels} * size_{filter})$ to induce a 4D tensor:

$$f_i(x, y, t) = \mathbf{u}_i(\mathbf{x}, \mathbf{y}, t) * \mathbf{I}(\mathbf{x}, \mathbf{y}, t) = \mathbf{u}_i(\mathbf{x}, \mathbf{y}) \cdot \mathbf{v}_j(t) * \mathbf{I}(\mathbf{x}, \mathbf{y}, t) \tag{2}$$

Onice, given the connection from matrix product to matrix convolution, if we can prove the optimality of deep 3D networks, intuitively we can adapt the proof to deep 3D CNNs.

Status Quo: 3D CNN is now Widely Used

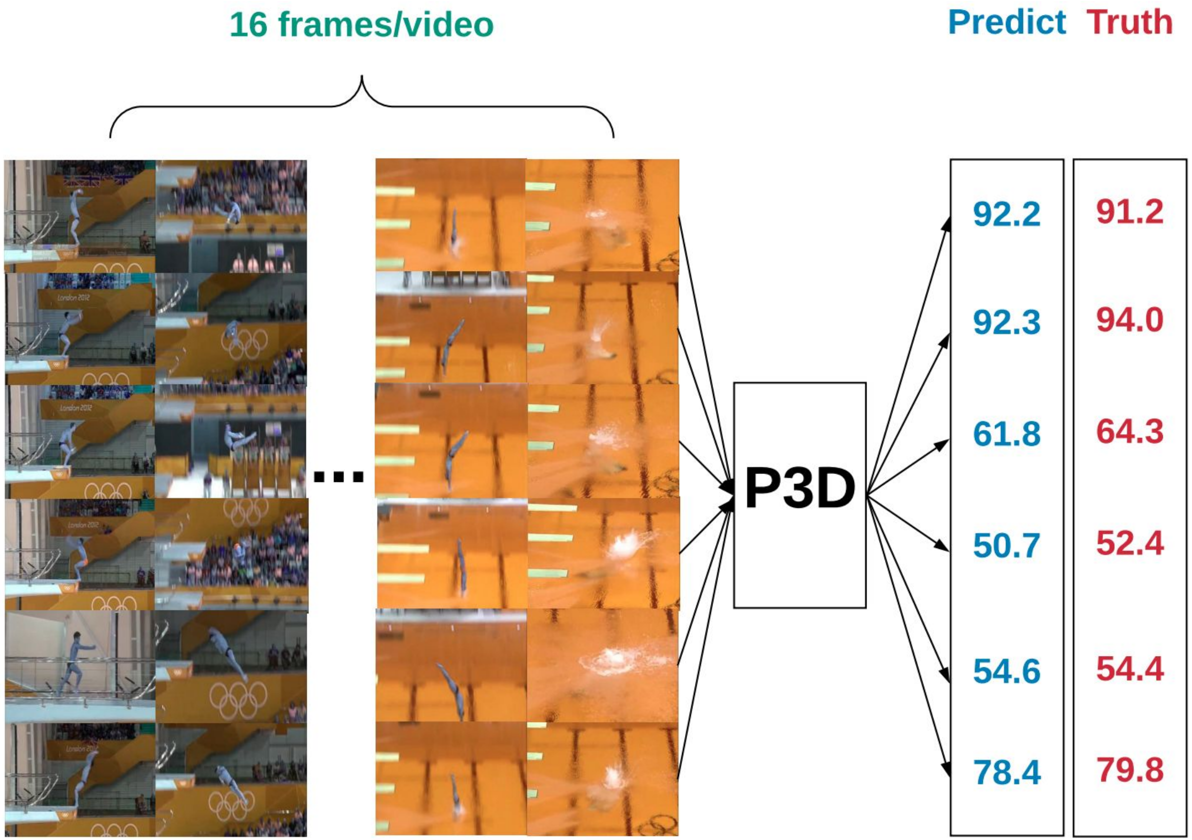


3D CNN is not particularly designed for video analysis. The patch-level 3D CNN has become a hard core formedical image analysis.

Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, 2017.
Kinetics forms the basis of an international human action classification competition being organised by **ActivityNet**.

Our First Attack

Adding a fully-connected layer on top of the 2nd last layers of P3D for regression. A training set of 16-frame clips sampled from raw videos of the UNLV-Diving dataset are input into the revised P3D network equipped with weights pre-trained on the Kinectics dataset. The scores predicted by the network (in blue) are compared with the ground truth in red.



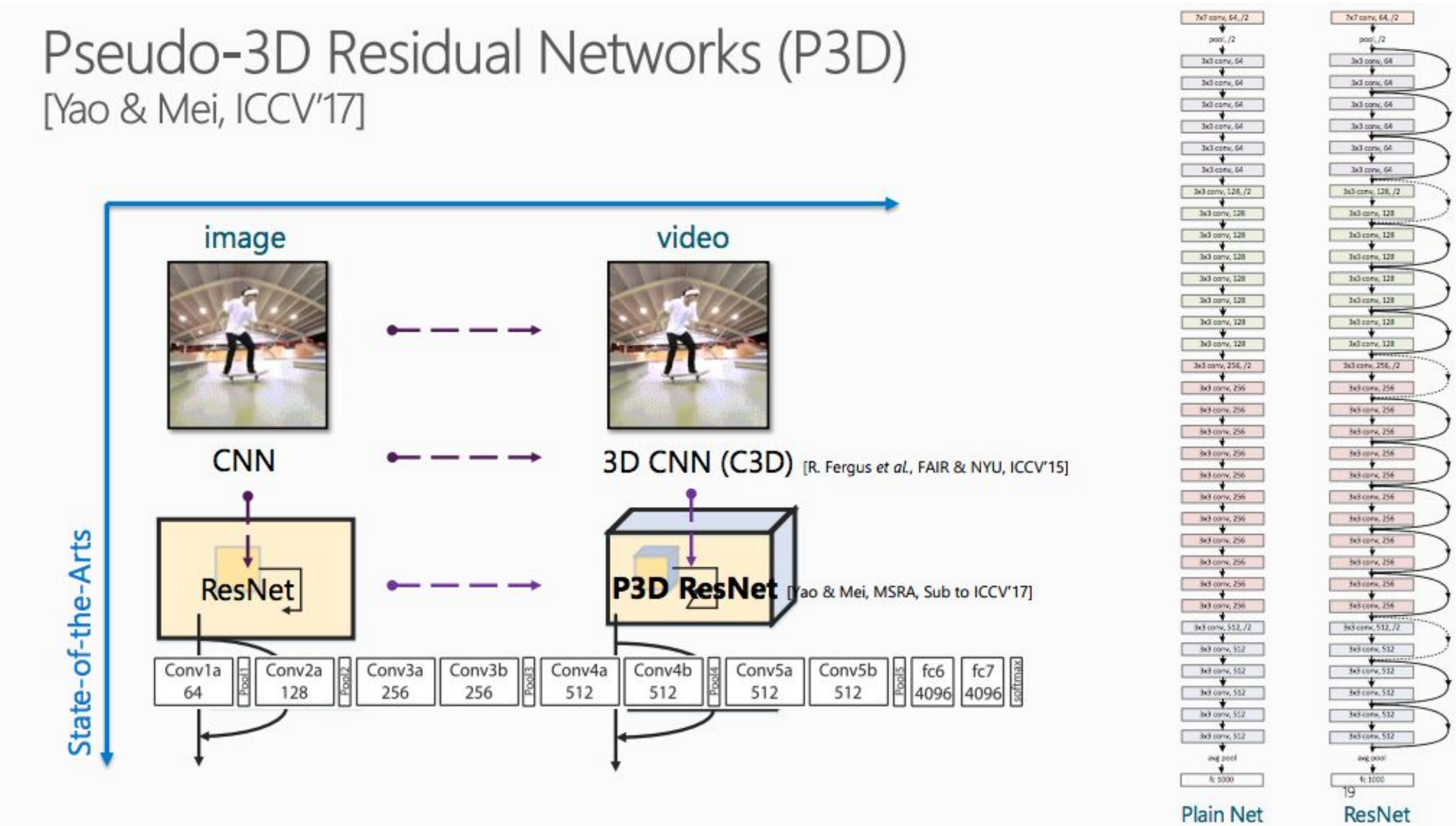
Our P3D-consecutive + FC regression, full video	0.43 ± 0.09
Our P3D-spaced + FC regression on full video	$\mathit{0.80} \pm 0.01$

Video Sampling

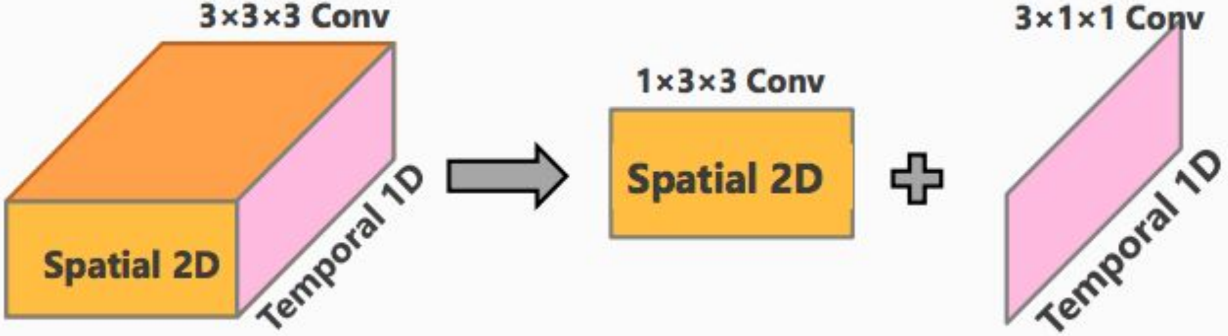
If a 3D CNN is trained for a preset number of frames, then it expects that number of frames at testing. The case is also true for fine-tuning pre-trained networks.

Video sampling. As P3D is designed to process clips of 16 frames, fine-turning the pre-trained P3D model needs clips of 16 frames. There are three effective strategies for sampling 16 frames from a video. In Sec. 4.3 we will compare them. Firstly, normally a 3D network needs to be trained for many epochs on small datasets. A good strategy turns out to be randomly stopping a sliding window of 16 consecutive frames along the temporal axis, which not only keeps the action smooth and coherent but also introduces certain randomness during each epoch. Thus, it also augments the data. We called models trained and tested using this strategy as *P3D-consecutive*. There is information loss during the random consecutive sampling. P3D can only read a 16-frame clip per video and thus can hardly see all the four stages relevant with scoring, while all influence the score. A bad case is that the first stage is sampled. Secondly, as video summarization, equally spaced sampling collectively represents the video and cover all stages, though sacrificing the temporal coherence. This model is called *P3D-spaced*.

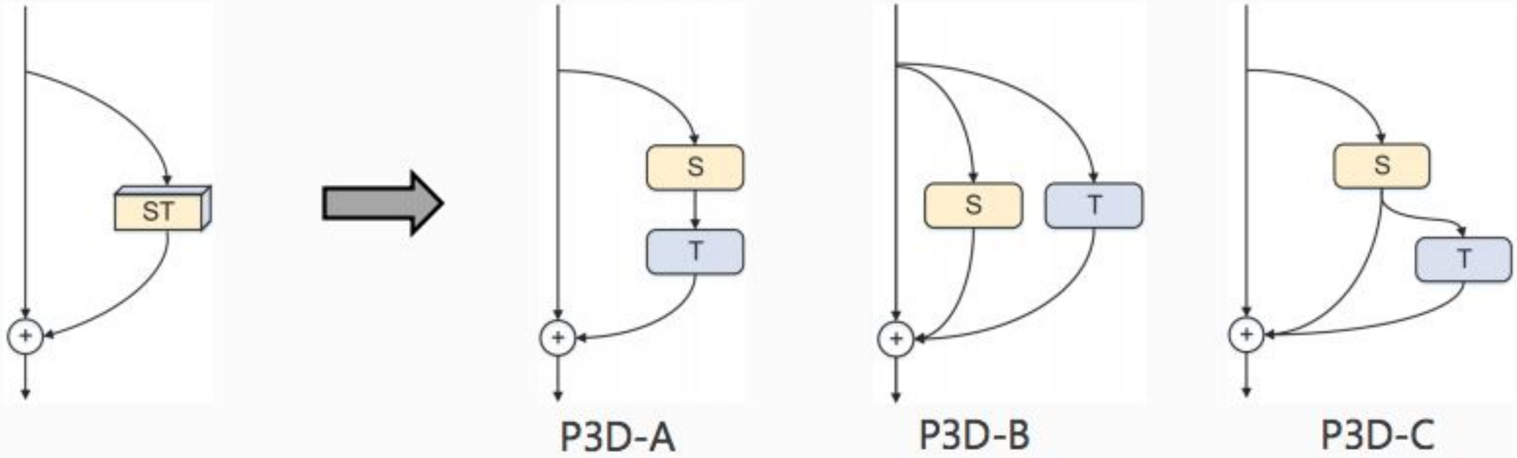
Existing Work: What's P3D?



Pseudo-3D Residual Networks (P3D) [Qiu, Yao, Mei, ICCV'17]



- Reduce model size
- Fully leverage pre-learned 2D CNN from image
- Enhance the structural diversity



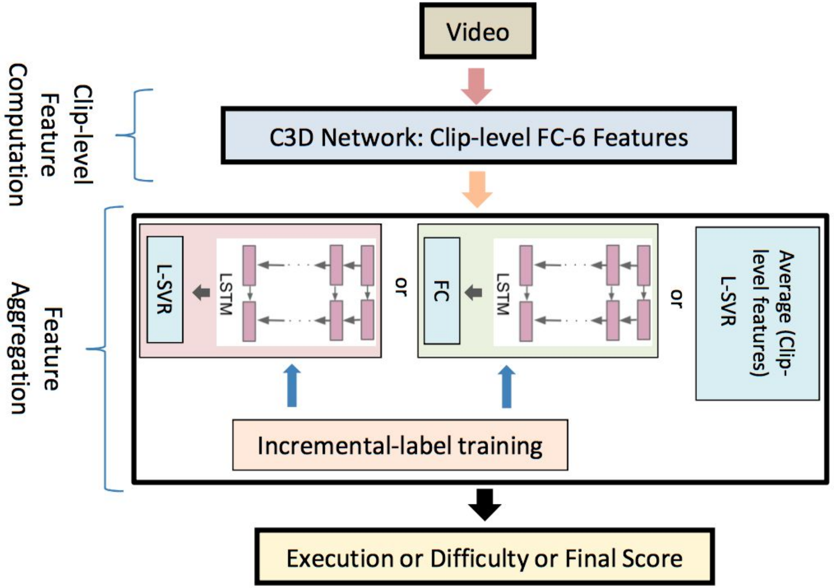
Our Implementation Details

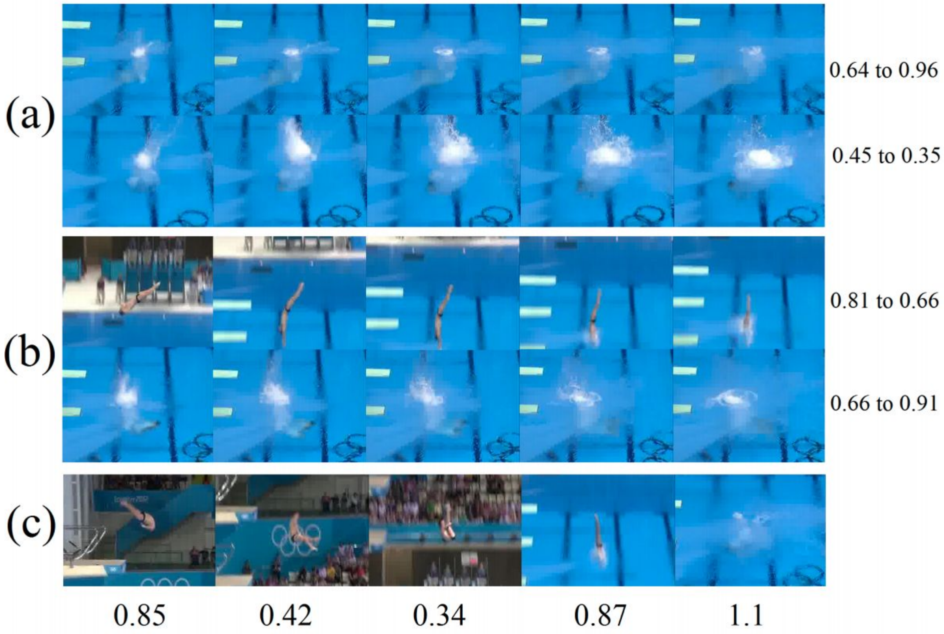
P3D. We use a PyTorch implementation of P3D-199 model[4] with weights pre-trained on Kinetics and revised it into a regression model. The *P3D-consecutive* model is trained and tested on 16 consecutive frames that randomly selected from the entire video. For the *P3D-spaced* model, frames are sampled from the video with equal space. The selected 16 frames are then resized into 160×160 to be input into models during training and testing. Residual units are in the order of P3D-serial P3D-parallel P3D-composition. Dropout with rate 0.5 is applied on the top FC layer. The MSE loss is used as the loss function in training. We use Adam with learning rate of 0.0001 as our optimizer. Models are trained for 90 epochs with learning decay factor of 0.1 for every 30 epochs.

Performance Comparison

Methods	Correlation
Hierarchical ConvISA [32] (ICCV 2011)	0.19
Pose+DCT+SVR (best in [4], ECCV 2014)	0.53
Entropy feature ApEnFT [6] (BMVC 2015)	0.45
C3D+LSTM [2] (CVPRW 2017)	0.36
C3D+LSTM+SVR [2] (CVPRW 2017)	0.66
C3D+SVR (the best in [2], CVPRW 2017)	0.74
Our P3D-consecutive + FC regression, full video	0.43 ± 0.09
Our P3D-spaced + FC regression on full video	$\mathit{0.80} \pm 0.01$

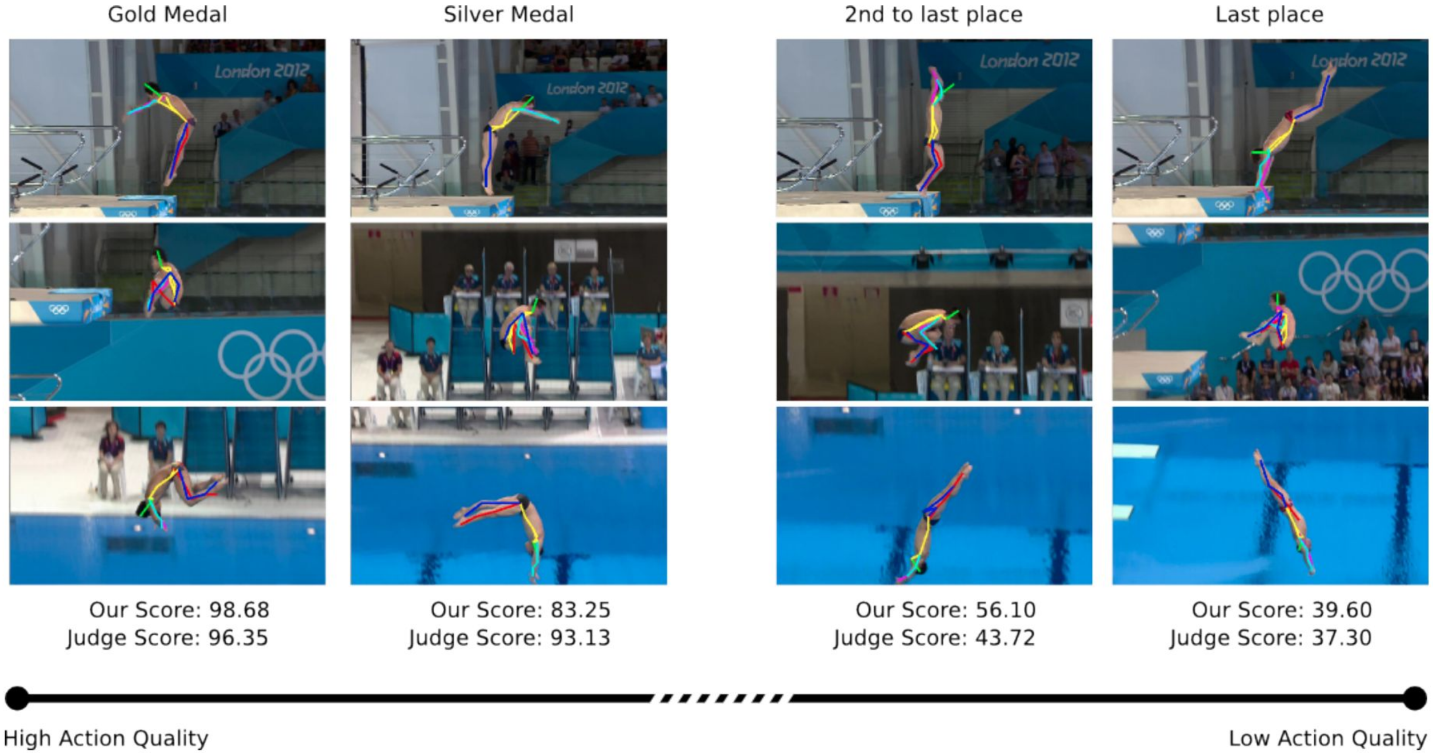
Exisiting Work [2]: Learning to score olympic events





[2] Parmar, P., Morris, B.: Learning to score olympic events. In CVPR 2017 Workshops.

Existing Work [4]: Assessing the quality of actions



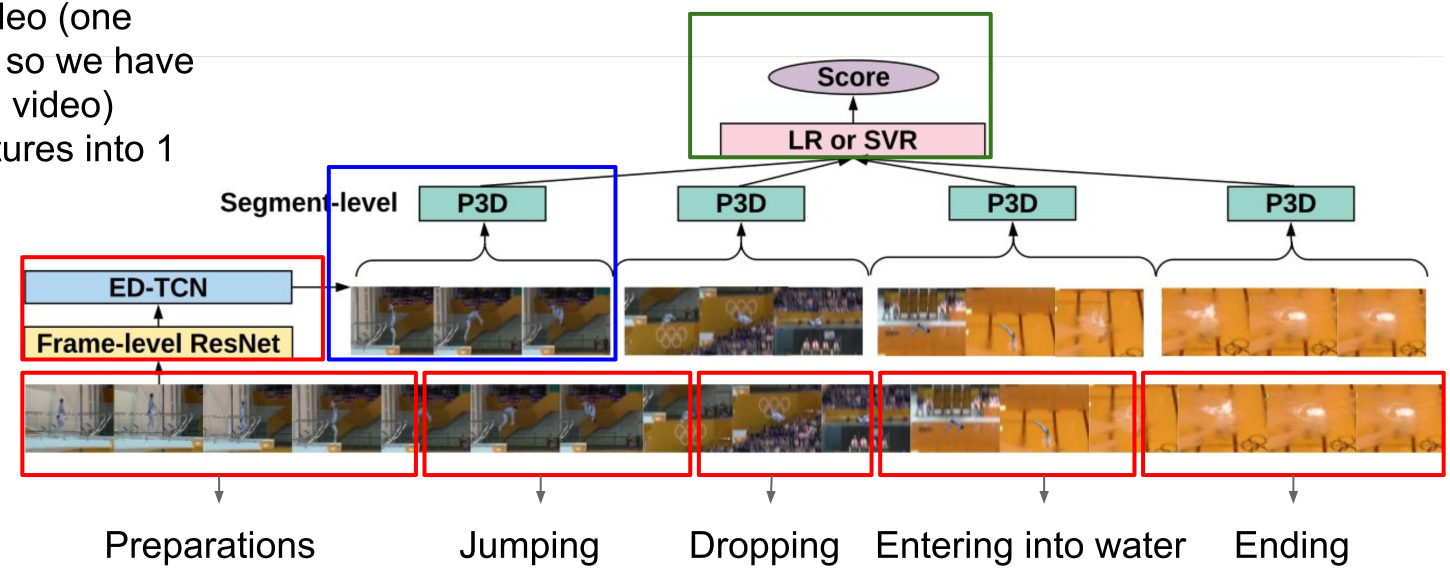
[4] Pirsiavash, H., Vondrick, C., Torralba, A.: Assessing the quality of actions. In ECCV 2014.

Proposing a Better Approach

- Use SVR ot LR to learn a mapping from the features to the score of videos

- Use four models to extract features for every video (one model on one stage, so we have 4 1-D feature for one video)
- Average these 4 features into 1

- On each stage (1-4), train a P3D independently



- Use TCN to segment tge video into five stages (0-4)

While our approach is not the first to score sports actions, to our knowledge it is the first to score them stage by stage.

Temporal segmentation using Tempoal CovNet (TCN)

The task of temporal segmentation in our case study is to classify frames into 5 classes with the intra-class continuity constraint. Taking the frame-level 2D CNN features as inputs, TCN can return five segments (one preparation stage, three action stages and one background stage) for a diving video. Suppose an input video has K frames and the output feature is D-dimensional, then the input to TCN can be denoted as $\mathbf{X}_0 \in \mathbb{R}^{D \times T}$ where the subscript is the count of layers traversed till now (below we use $l = 0, 1, \ldots$ for layer). For the ED-TCN [13], the temporal convolution is represented as

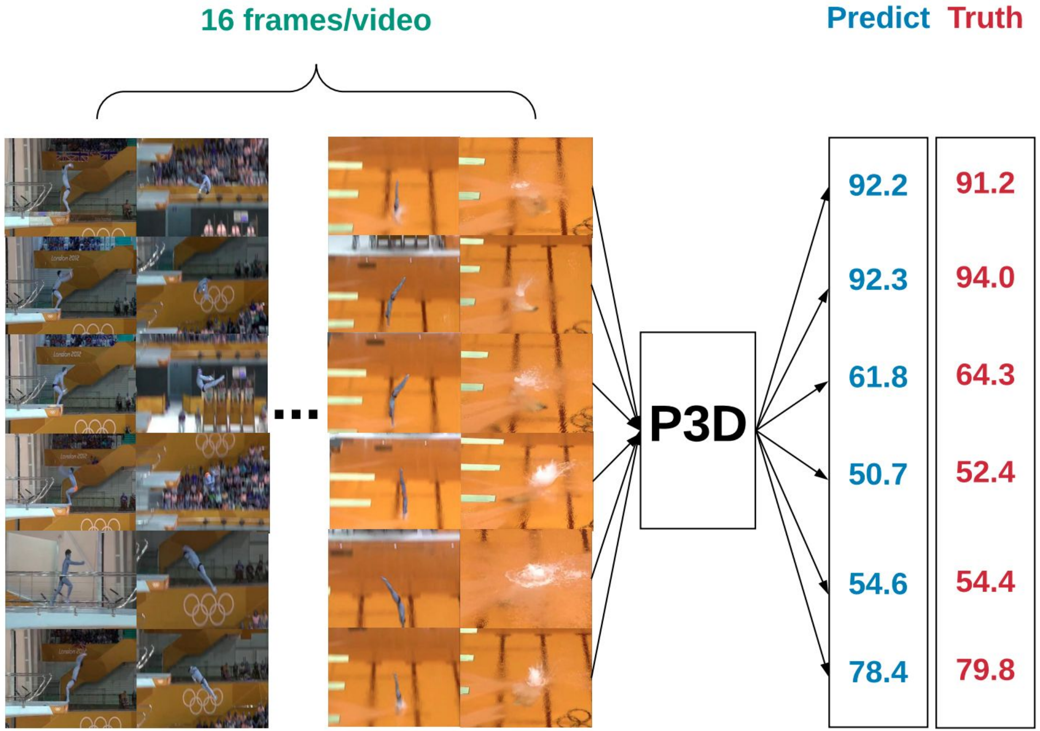
$$\mathbf{X}_l = f(\mathbf{W}_l * \mathbf{X}_{l-1} + \mathbf{b}) \tag{6}$$

where $\mathbf{X}_l \in \mathbb{R}^{N_l \times T_l}$, $N_0 = D$, $T_0 = K$. The convolution filters are parameterized by $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{N_l}$, $\mathbf{w}_i \in \mathbb{R}^{d_l \times N_{l-1}}$ and $b \in \mathbb{R}^{N_l}$ for N_l being the number of convolution filters at l-th layer, T_l being the number of features, d_l being the filter length at l-th layer and $f(\cdot)$ being the activation function.

Colin Lea, Michael Flynn, Rene Vidal, Austin Reiter, Gregory D. Hager. Temporal Convolutional Networks for Action Segmentation and Detection. CVPR 2017.

What's the Difference?

Authors of [4] run SVR on human poses to score their MIT Diving dataset upon which authors of [2] build the UNLV-Dive dataset. The C3D+SVR approach of [2] has shown significant improvements over previous works [6] using the approximate entropy features.



[2] Parmar, P., Morris, B.: Learning to score olympic events. In CVPR 2017 Workshops.
[4] Pirsiavash, H., Vondrick, C., Torralba, A.: Assessing the quality of actions. In ECCV 2014.
[6] Venkataraman, V., Vlachos, I., Turaga, P.: Dynamical regularity for action analysis. In BMVC 2015.

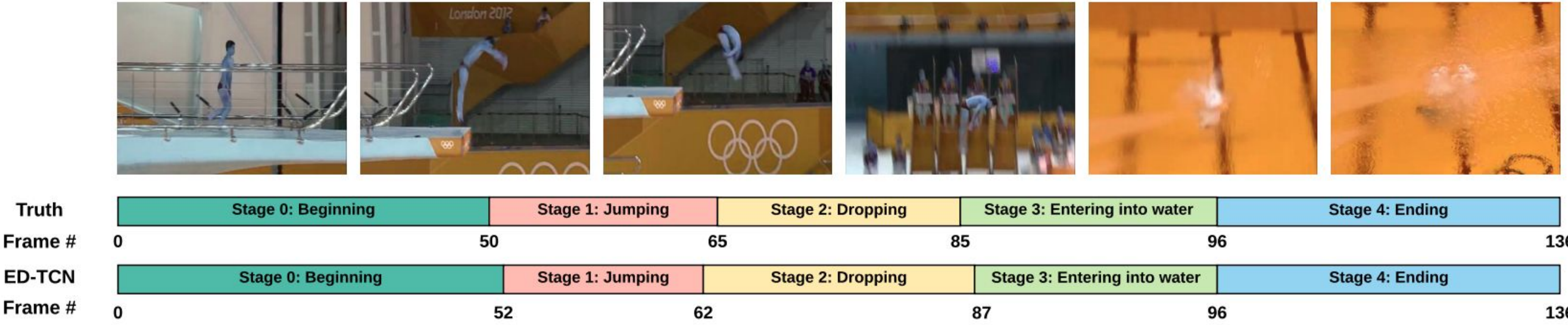
Performance Again!

Per-Stage Sampling P3D-center. If the center frame of each stage is given, we choose the P3D input to be the 16 frames that are centered at the middle of each stage and have a spacing of 1. They serve as a summarization of this stage and will be input into the P3D-center model of that stage.

Methods	Correlation
Hierarchical ConvISA [18] (ICCV 2011)	0.19
Pose+DCT+SVR (best in [1], ECCV 2014)	0.53
Entropy feature ApEnFT [3] (BMVC 2015)	0.45
C3D+LSTM [2] (CVPRW 2017)	0.36
C3D+LSTM+SVR [2] (CVPRW 2017)	0.66
C3D+SVR (the best in [2], CVPRW 2017)	0.74
Our P3D-consecutive + FC regression, full video	0.43 ± 0.09
Our P3D-spaced + FC regression on full video	$\mathit{0.80} \pm 0.01$
Our P3D-center + FC regression, jumping stage	0.49 ± 0.04
Our P3D-center + FC regression, dropping stage	0.60 ± 0.03
Our P3D-center-FC, jumping-dropping combined	0.47 ± 0.04
Our P3D-center + FC on entering into water stage	$\mathit{0.82} \pm 0.01$
Our P3D-center + FC, videos except ending stage	0.56 ± 0.04
Our P3D-center + FC regression, ending stage	$\mathit{0.77} \pm 0.02$
LR on scores output by stage-wise P3D-center-FC	*0.82*
SVR on score output by stage-wise P3D-center-FC	*0.84*
LR on average of stage-wise P3D-center features	*0.81*
SVR on average of stage-wise P3D-center features	**0.86**
Concatenation of stage-wise P3D-center features	**0.86**

Table 1. Pearson correlation comparison on official split-4.

Visualization of TCN segmentation Intermediate result



Temporal model	Acc (%)
Bi-LSTM [33]	95.7
ED-TCN	**96.6**
Tricornet (TCN+Bi-LSTM) [34]	96.0

Table 2. Accuracy comparison of temporal classification.

THANKS!

Questions and suggestions!

Contact: xxiang@cs.jhu.edu