Revisiting Skeleton-based Action Recognition

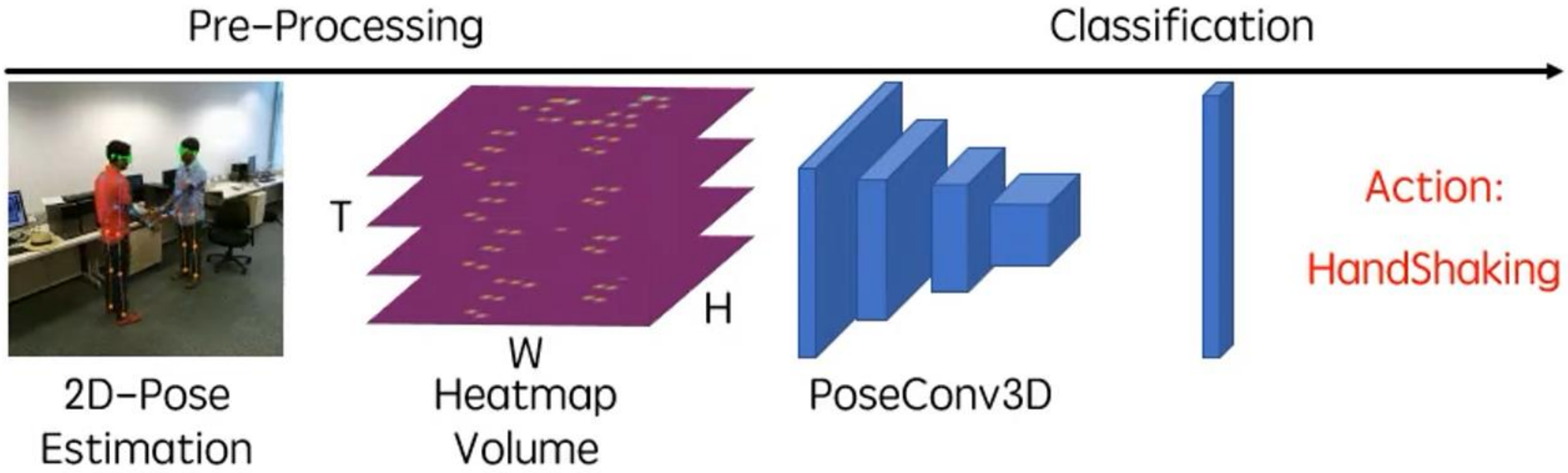
Haodong Duan et al. (CVPR 2022)

발표자: 손소영

Summary

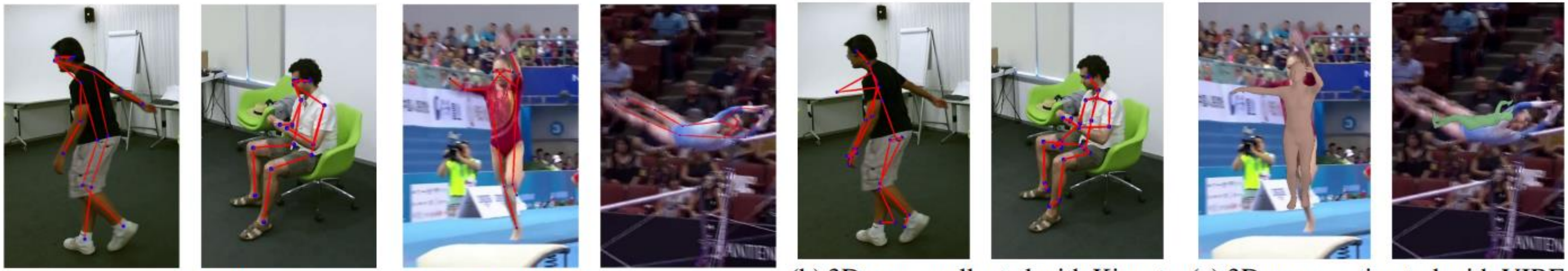
- A new approach to skeleton-based action recognition, using 3D-CNN instead of GCN
- Outperforming GCN under various settings with improved robustness, interoperability, and scalability
- Handle multi-person scenarios without additional computation costs
- Easily integrated with other modalities at early fusion stages
- Achieved SOTA on 5 of 6 standard skeleton-based action recognition
- Achieved SOTA on all 8 multi-modality action recognition benchmarks

Pipeline



1. Use a two-stage pose estimator(detection + pose estimation) for 2D HPE
2. Stack heatmaps of joints or limbs along the temporal dimension and apply pre-processing to the generated 3D heatmap volumes
3. Use a 3D-CNN to classify the 3D heatmap volumes

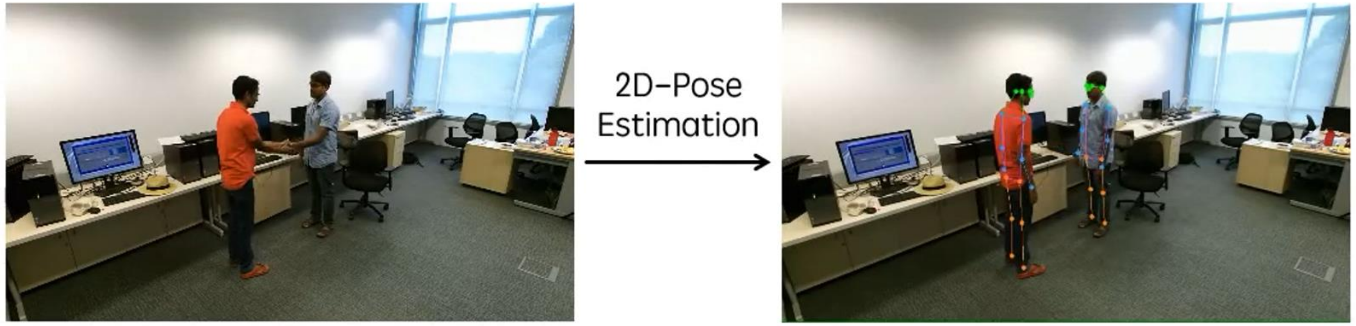
2D Pose Estimator



(a) 2D poses estimated with HRNet. (b) 3D poses collected with Kinect. (c) 3D poses estimated with VIBE.

- Better quality with 2D poses comparing to 3D poses
- Adopt 2D top-down pose estimators for pose extraction

2D Pose Estimator

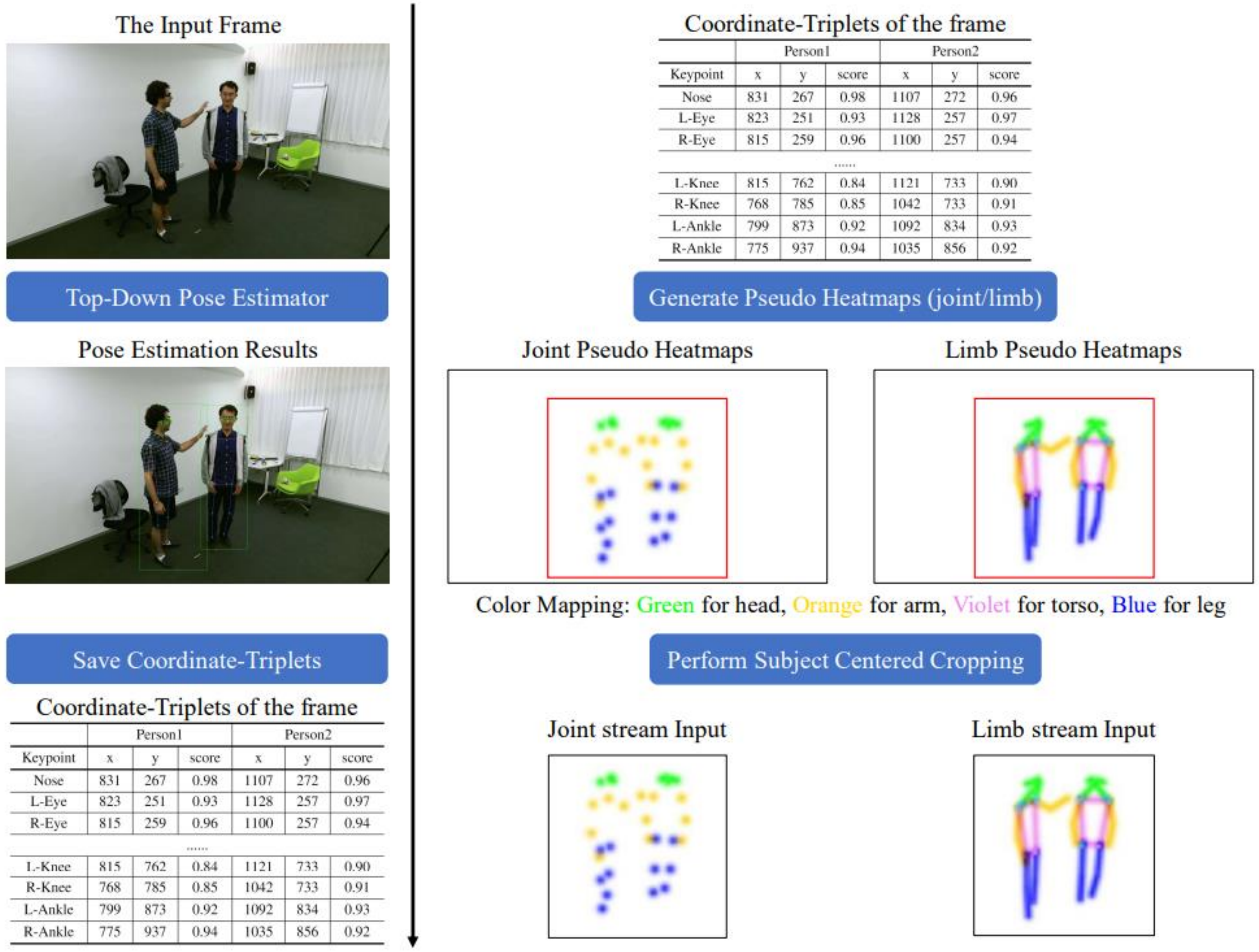


Person 1	Person 2
Left-shoulder (x_{11}, y_{11}, c_{11})	Left-shoulder (x_{21}, y_{21}, c_{21})
Right-shoulder (x_{12}, y_{12}, c_{12})	Right-shoulder (x_{22}, y_{22}, c_{22})
......	......
Right-ankle (x_{1k}, y_{1k}, c_{1k})	Right-ankle (x_{2k}, y_{2k}, c_{2k})

- Estimated heatmaps are stored as coordinate-triplets (x, y, c)

  c : maximum score of the heatmap (x, y): coordinate of c
- Coordinate-triplets help save most storage space at the cost of little performance drop

Convert to 3D Heatmap Volumes



Coordinate-Triplets of the frame

	Person1			Person2		
Keypoint	x	y	score	x	y	score
Nose	831	267	0.98	1107	272	0.96
L-Eye	823	251	0.93	1128	257	0.97
R-Eye	815	259	0.96	1100	257	0.94
......						
L-Knee	815	762	0.84	1121	733	0.90
R-Knee	768	785	0.85	1042	733	0.91
L-Ankle	799	873	0.92	1092	834	0.93
R-Ankle	775	937	0.94	1035	856	0.92

- Represent 2D pose as a heatmap of size $K \times H \times W$

  (K: # of joints, H, W: height and width of the frame)

- Can directly use the heatmap produced by the Top-Down pose estimator as the target heatmap

Convert to 3D Heatmap Volumes

- Coordinate-triplets -> joint heatmap J by composing K Gaussian maps centered at every joint:

$$\boldsymbol{J}_{kij} = e^{-\frac{(i-x_k)^2+(j-y_k)^2}{2*\sigma^2}} * c_k$$

σ: controls the variance of gaussian maps
(x_k, y_k): the location of the k-th joint
c_k: the confidence score of the k-th joint

- Limb heatmap L

$$\boldsymbol{L}_{kij} = e^{-\frac{\mathcal{D}((i,j),seg[a_k,b_k])^2}{2*\sigma^2}} * \min(c_{a_k}, c_{b_k})$$

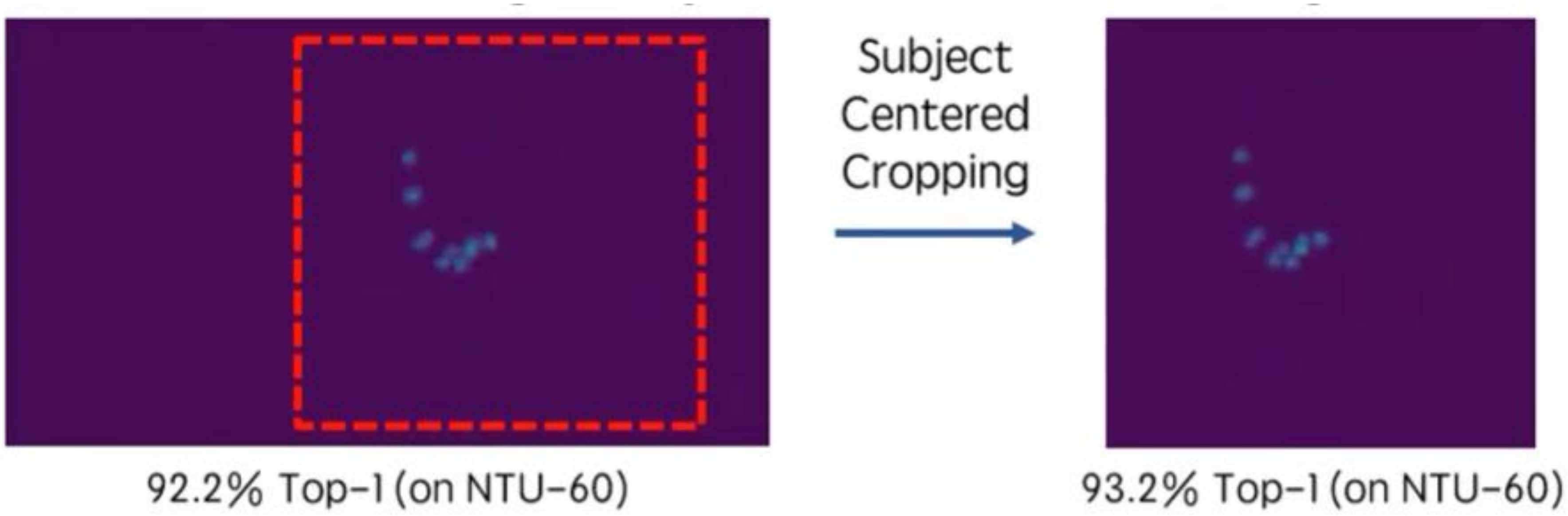
k-th limb: between two joints a_k and b_k
$\mathcal{D}$: distance from the point (i, j) to the segment $[(x_{a_k}, y_{a_k}), (x_{b_k}, y_{b_k})]$

Convert to 3D Heatmap Volumes

- Easily extend it to the multi-person case by accumulating k-th gaussian maps of all persons without enlarging the heatmap
- Obtain 3D heatmap volume by stacking all heatmaps (J or L) along the temporal dimension

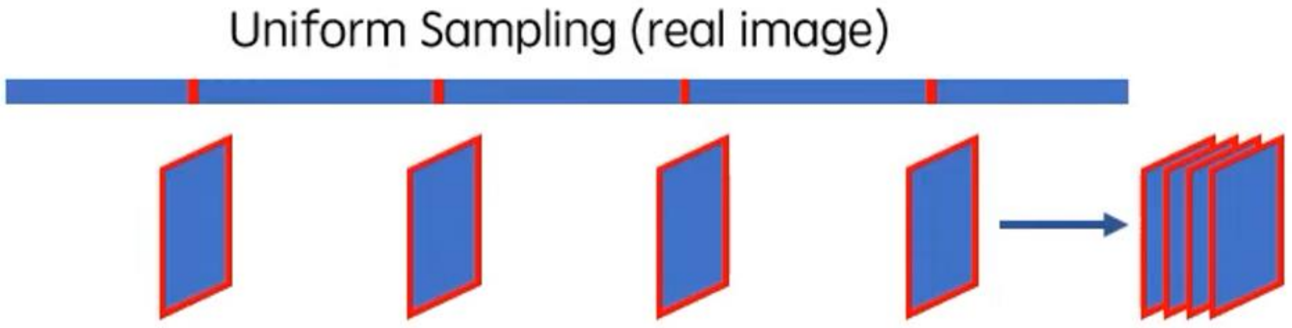
  -> size of $(K \times T \times H \times W)$

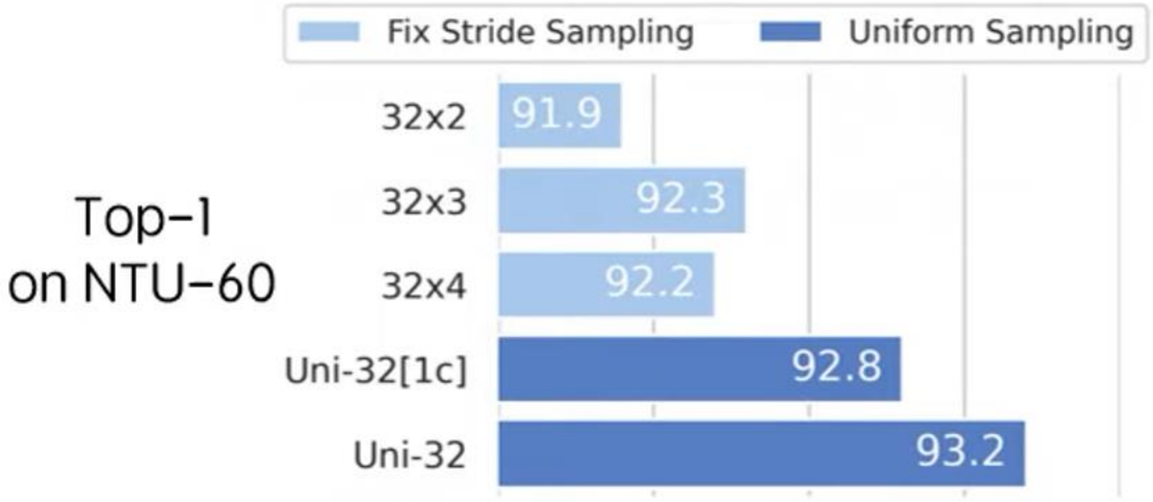
Two techniques to reduce the redundancy of 3D heatmap volumes



1. Subjects-Centered Cropping
   - To reduce spatial redundancy
   - Find the smallest bounding box that envelops all the 2D poses across frames
   - Crop all frames according to the found box and resize them to the target size

Two techniques to reduce the redundancy of 3D heatmap volumes





2. Uniform Sampling
   - To reduce temporal redundancy
   - To sample n frames from a video, divide the video into n segments of equal length and randomly select one frame from each segment
   - Better at maintaining the global dynamics of the video

3D-CNN as backbone

- Design two families of 3D-CNNs:
  - PoseConv3D for the Pose modality
  - RGBPose-Conv3D for the RGB+pose dual modality
- Demonstrates the power of 3D-CNN in capturing spatiotemporal dynamics of skeleton sequences

PoseConv3D

- Focuses on human skeletons
- Takes 3D heatmap volumes as input
- Can be instantiated with various 3D-CNN backbones
- Two modifications are needed to adapt:
  - Down-sampling in early stages are removed from the 3D-CNN
    - Since the spatial resolution of 3D heatmap volumes does not need to be as large as RGB clips
  - Shallower and thinner network is sufficient
    - 3D heatmap volumes are already mid-level features for action recognition
- Three popular 3D-CNNS: C3D, SlowOnly(default), X3D

C3D

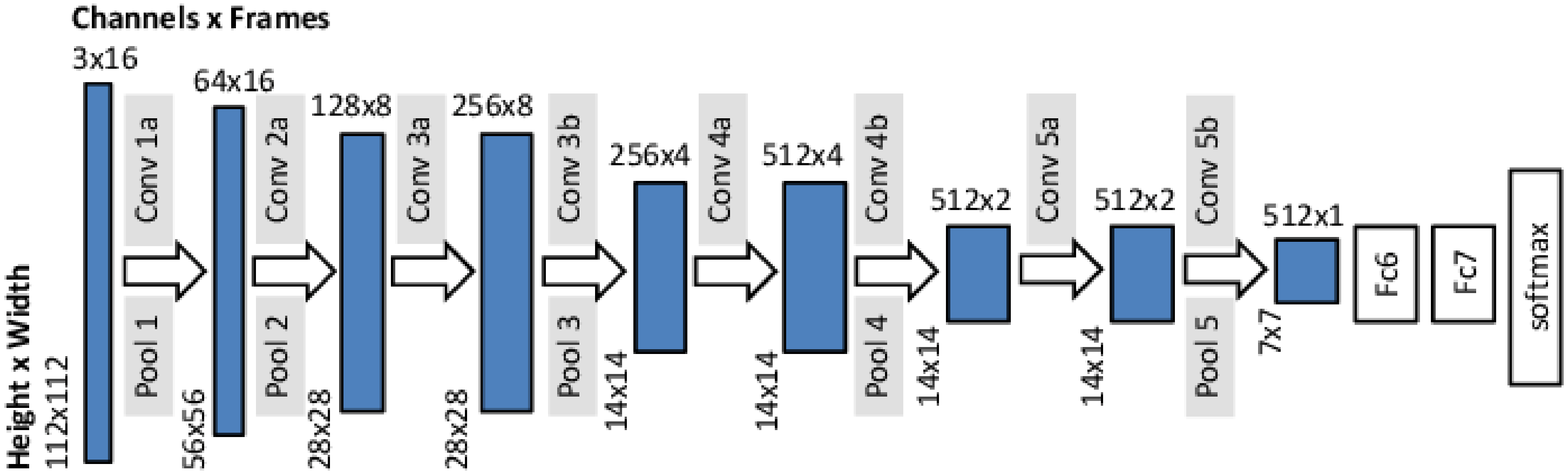
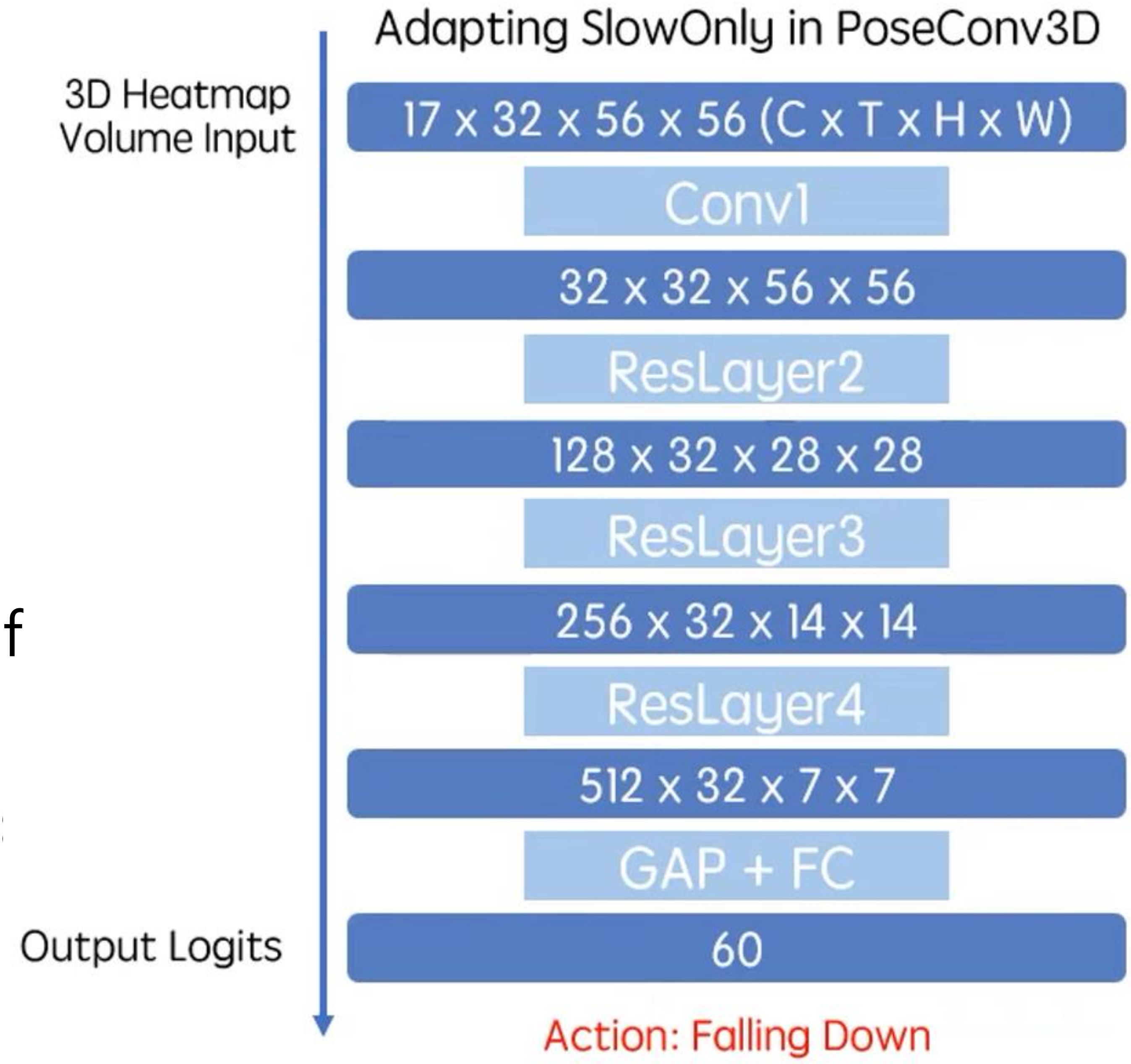


Fig. 3: C3D architecture with eight convolution layers, five max pooling layers and two fully connected layers.

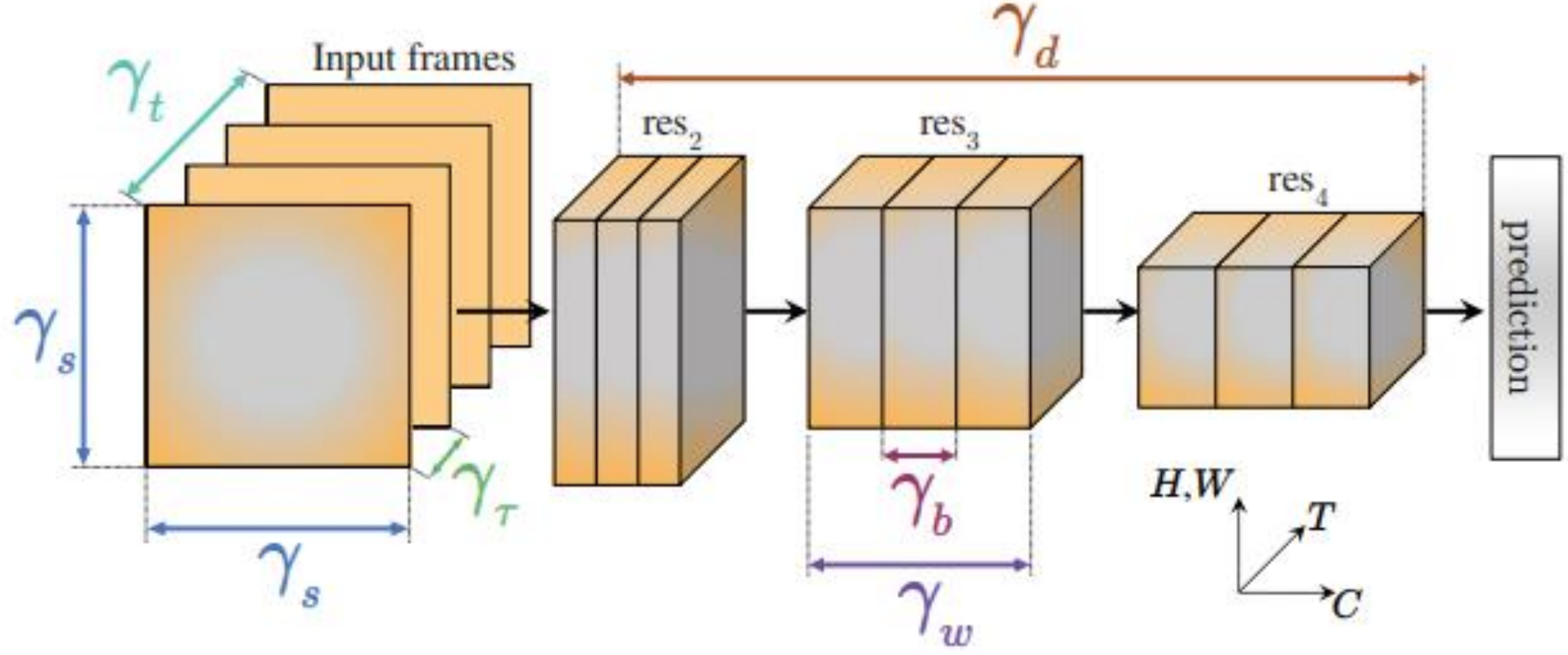
- One of the earliest 3D-CNN model for RGB-based action recognition
- Reduce its channel-width to half(64 -> 32) for better efficiency
- Remove last two convolution layers

SlowOnly(default)

- Obtained by inflating the ResNet layers in the last two stages from 2D to 3D
- Reduce channel-width to half (64 -> 32) as well as remove the original first stage in the network



X3D

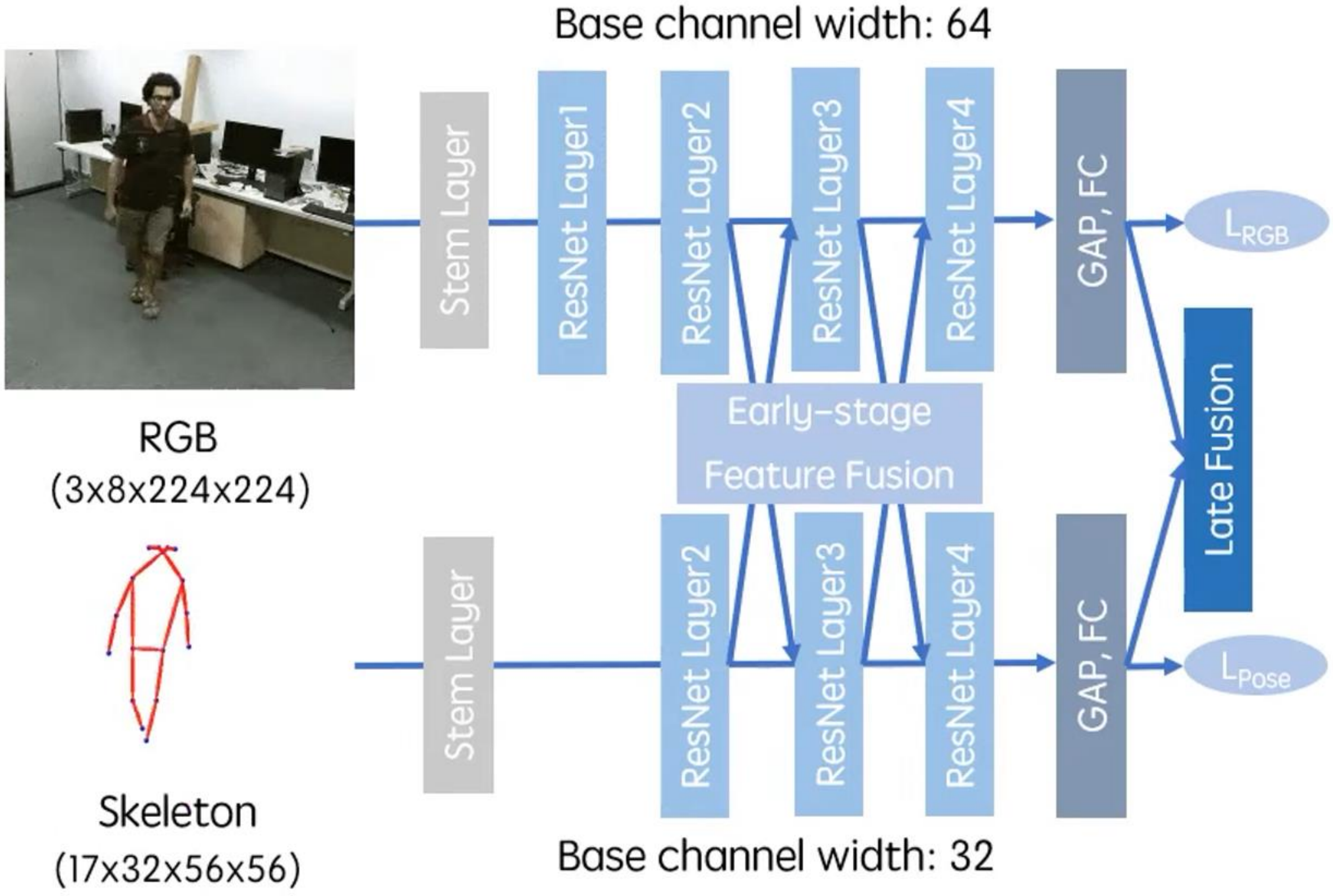


- Recent SOTA 3D-CNN model for action recognition
- Replace vanilla convolutions with depth-wise convolutions
- Competitive performance with tiny amount of parameters

RGBPose-Conv3D

- Early fusion of human skeletons and RGB frames
- Two-stream 3D-CNN with two pathways that respectively process RGB modality and Pose modality
- Two pathways are asymmetrical due to the different characteristics of the two modalities
  - Pose pathway has a smaller channel width, a smaller depth as well as a smaller input spatial resolution, compare to RGB
  - Bidirectional connections between the two pathways are added to promote early-stage feature fusion between two modalities
- Trained with two individual cross-entropy losses

RGBPose-Conv3D



Performances

Skeleton Based Action Recognition	Kinetics-Skeleton dataset	PoseC3D	Accuracy	47.7	# 2	Compare
Skeleton Based Action Recognition	Kinetics-Skeleton dataset	PoseC3D (SlowOnly-346)	Accuracy	49.1	# 1	Compare
Skeleton Based Action Recognition	NTU RGB+D	PoseC3D [3D Heatmap]	Accuracy (CV)	97.1	# 6	Compare
			Accuracy (CS)	94.1	# 1	Compare
Action Recognition	NTU RGB+D	PoseC3D (RGB + Pose)	Accuracy (CS)	97.0	# 1	Compare
			Accuracy (CV)	99.6	# 1	Compare
Action Recognition	NTU RGB+D 120	PoseC3D (RGB + Pose)	Accuracy (Cross-Subject)	96.4	# 1	Compare
			Accuracy (Cross-Setup)	95.3	# 1	Compare
Skeleton Based Action Recognition	NTU RGB+D 120	PoseC3D (w. HRNet 2D Skeleton)	Accuracy (Cross-Subject)	86.9	# 17	Compare
			Accuracy (Cross-Setup)	90.3	# 10	Compare
Group Activity Recognition	Volleyball	PoseC3D (Pose-Only)	Accuracy	91.3	# 7	Compare
Action Recognition	Volleyball	PoseC3D (Pose Only)	Accuracy	91.3	# 1	Compare

Performances

	MS-G3D			Pose-SlowOnly			
Dataset	Acc	Params	FLOPs	1-clip	10-clip	Params	FLOPs
FineGYM	92.0	2.8M	24.7G	**92.4**	**93.2**	**2.0M**	**15.9G**
NTU-60	91.9	2.8M	16.7G	**93.1**	**93.7**		
NTU-120	84.8	2.8M	16.7G	**85.1**	**86.0**		
Kinetics400	**44.9**	2.8M	17.5G	44.8	**46.0**		

GCN vs. PoseConv3D

	late fusion	RGB → Pose	Pose → RGB	RGB ↔ Pose
1-clip	92.6	93.0	93.4	**93.6**
10-clip	93.4	93.7	93.8	**94.1**

The design of RGBPose-Conv3D

	RGB	Pose	late fusion	early+late fusion
FineGYM	87.2 / 88.5	91.0 / 92.0	92.6 / 93.4	**93.6 / 94.1**
NTU-60	94.1 / 94.9	92.8 / 93.2	95.5 / 96.0	**96.2 / 96.5**

The universality of RGBPose-Conv3D

Performances

Method	NTU60-XSub	NTU60-XView	NTU120-XSub	NTU120-XSet	Kinetics	FineGYM
ST-GCN [71]	81.5	88.3	70.7	73.2	30.7	25.2*
AS-GCN [34]	86.8	94.2	78.3	79.8	34.8	-
RA-GCN [54]	87.3	93.6	81.1	82.7	-	-
AGCN [51]	88.5	95.1	-	-	36.1	-
DGNN [50]	89.9	96.1	-	-	36.9	-
FGCN [72]	90.2	96.3	85.4	87.4	-	-
Shift-GCN [9]	90.7	96.5	85.9	87.6	-	-
DSTA-Net [52]	91.5	96.4	86.6	89.0	-	-
MS-G3D [40]	91.5	96.2	86.9	88.4	38.0	-
MS-G3D ++	92.2	96.6	**87.2**	89.0	45.1	92.6
PoseConv3D ($\boldsymbol{J}$)	**93.7**	**96.6**	86.0	**89.6**	**46.0**	**93.2**
PoseConv3D ($\boldsymbol{J} + \boldsymbol{L}$)	**94.1**	**97.1**	86.9	**90.3**	**47.7**	**94.3**

Compare to previous SOTA models

Performances

(a) Mulit-modality action recognition with *RGBPose-Conv3D*.

Dataset	Previous state-of-the-art	**Ours**
FineGYM-99	87.7 (R) [30]	**95.6** (R + P)
NTU60 (X-Sub / X-View)	95.7 / 98.9 (R + P) [14]	**97.0 / 99.6** (R + P)
NTU120 (X-Sub / X-Set)	90.7 / 92.5 (R + P) [12]	**95.3 / 96.4** (R + P)

(b) Mulit-modality action recognition with *LateFusion*.[5]

Dataset	Previous state-of-the-art	**Ours (Pose)**	**Ours (Fused)**
Kinetics400	84.9 (R) [39]	47.7	**85.5** (R + P)
UCF101	98.6 (R + F) [16]	87.0	**98.8** (R + F + P)
HMDB51	83.8 (R + F) [16]	69.3	**85.0** (R + F + P)

Compare to previous SOTA models

References

Presentation video: https://www.youtube.com/watch?v=OFDv5hvq_7s

Paper: https://arxiv.org/pdf/2104.13586v2.pdf

Github: https://github.com/kennymckormick/pyskl

Thank you