Unified Keypoint-based Action Recognition Framework via Structured Keypoint Pooling Supplementary Material

Ryo Hachiuma[*], Fumiaki Sato[*], Taiki Sekii
Konica Minolta, Inc.
{rhachiuma,fumiaki.sato.jp,taiki.sekii}@gmail.com

1. Implementation Details

In this section, we provide detailed information about the proposed network architecture, data augmentation, hyperparameters employed during training, and preprocessing.

1.1. Network Architecture

The dimension of the output feature vector at the Point embedding layer is set to 256, and the dimensions of the input feature vector at the MLP Blocks are set to $256, 512$, and 1024, respectively, as the Grouped Pool Block doubles the channels via concatenation. The number of repeats in the MLP Block is set to $r = 2$, and the MLP expansion ratio is set to $\alpha = 2.0$. We employ Batch Normalization [3] as the normalization layer and GELU [2] as the activation function in the MLP Block. We implement FC layers using a stack of three MLP layers (1024, 512, and C) with a ReLU activation function [6] and a Batch Normalization layer [3], where C denotes the number of classes in the dataset. We also apply a Droppath layer [4] with a probability of 0.1 during training to each residual module in the MLP Block to achieve stable training.

1.2. Data Augmentation

We apply two types of data augmentation during training; augmentation onto the image space and along the temporal axis to the input keypoints. We randomly scale, shift, rotate, and flip the keypoint coordinates for the augmentation onto the image space. We randomly crop the input keypoints with a random size of the temporal window and a random start time. We also drop keypoints within a random interval range.

1.3. Hyperparameters

The hyperparameters employed in each experiment are listed in Tab. 1. They were simply found in a standard coarse-to-fine grid search or step-by-step tuning. Since we evaluate the accuracy of the model trained with the Kinetics-400 dataset for the Mimetics experiment (Sec. 4.6), no hyperparameters are used for training on the Mimetics dataset. In the Kinetics-400 dataset, we employ the same hyperparameters to train different inputs such as a bone, a skeleton and an object, or skeletons detected by the HRNet or PPNv2.

1.4. Preprocessing of the Input Keypoints

In Sec 3.1 in the paper, we mentioned the number of input points is FIK, where F denotes the number of frames in the video clip, I denotes the number of instances per frame, and K denotes the number of keypoints per instance. When both human and object keypoints are input, the number of points is $F(I_oK_o + I_hK_h)$. I_o, I_h are the number of instances per frame; K_o, K_h are the number of keypoints per instance for objects and humans, respectively.

These parameters, F, I, K, are not hyperparameters for each dataset; an arbitrary number of points can be input due to the permutation-invariant property (theoretical) of the model. To simplify the implementation, F, I, K are fixed by setting the maximum values for each dataset. If the number of detected points is lower than the fixed value, an input tensor is padded with zero for each video. F is set as the maximum frame length for each dataset. I_o and I_h are fixed as the maximum detected instances used in the previous studies. K_o and K_h are fixed as the keypoint-detector settings. The undetected keypoints are padded with zero.

2. Further Results

2.1. Effectiveness of the Joint-Bone Ensemble

Tab. 2 summarizes the effectiveness of the joint-bone ensemble framework by employing the Kinetics-400 dataset. In this framework [1, 5], a separated model with an identical architecture is trained using the vector differences of the adjacent joints (bones) in each skeleton as an input, instead of the keypoint coordinates. The softmax scores of the joint

[*] Equal contribution.

Table 1. Hyperparameters of each dataset during training.

	Kinetics-400	RWF-2000	Hockey-Fight	Crowd Violence	Movies-Fight	UCF101	HMDB51	Mixamo	UCF101-24
Section	4.4, 4.5, 4.7	4.6	4.6	4.6	4.6	4.6	4.6	4.8	4.9
Optimizer	Stocastic Gradient Descent								
Number of epochs	150	60	30	30	30	100	20	60	60
Batch size	120	80	40	40	40	120	60	120	30
Learning rate	0.12	0.06	0.04	0.04	0.04	0.03	0.01	0.12	0.0075
Weight decay	0.00005	0.0002	0.0016	0.0016	0.0016	0.000025	0.0002	0.00005	0.000025
momentum	0.9								
LR scheduler	linear								
pretraining dataset	None	Kinetics-400						None	Kinetics-400
keypoint scaling	$[0.8, 1.2]$	$[0.4, 1.6]$	$[0.7, 1.3]$	$[0.6, 1.4]$	$[0.7, 1.3]$	$[0.5, 1.5]$	$[0.5, 1.5]$	$[1.0, 2.5]$	$[0.7, 1.3]$
keypoint shift	0.2	0.5	0.3	0.4	0.3	0.5	0.5	0.6	0.2
keypoint rotate (°)	10.0	30.0	20.0	30.0	20.0	10.0	15.0	30.0	5.0
keypoint flip ratio	0.5								
temporal crop window	100	150	150	200	100	150	150	100	200
temporal shift range	150	75	0	0	0	150	150	150	150
temporal FPS drop	5	5	3	3	3	3	3	5	1

Table 2. Ablation study of the joint-bone ensemble by employing the Kinetics-400 dataset.

Keypoint	Skeleton		Object		Acc. (%)
	Joint	Bone	Joint	Bone	
HRNet	✓				48.5
		✓			46.7
	✓	✓			50.3
PPNv2	✓				41.0
		✓			39.3
	✓	✓			43.1
	✓		✓		49.2
		✓		✓	50.1
	✓	✓	✓	✓	52.3

and bone models are summed to obtain the final prediction scores. Regarding the object contour points, we calculate the vector differences of adjacent keypoints such as *up left* and *up* or *right bottom* and *right*. Tab. 2 shows that the ensemble framework improves the recognition accuracy using the proposed point-cloud-based DNN. Also, this ensemble is effective not only for the skeleton input but also for the skeleton and object input.

References

[1] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting Skeleton-based Action Recognition. In *CVPR*, 2022. 1

[2] Dan Hendrycks and Kevin Gimpel. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415, 2016. 1

[3] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015. 1

[4] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. FractalNet: Ultra-Deep Neural Networks without Residuals. In *ICLR*, 2017. 1

[5] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In *CVPR*, 2020. 1

[6] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, 2010. 1