3D Pose Motion Representation for Action Recognition

Shengyu Huang
D-BAUG, ETH Zurich
shenhuan@student.ethz.ch

Ye Hong
D-BAUG, ETH Zurich
hongy@student.ethz.ch

Jingtong Li
D-BAUG, ETH Zurich
lijing@student.ethz.ch

Abstract

In this report, we study the task of action recognition from video clips. Most top-performing approaches[5, 26] follow a two-stream architecture that deals with appearance and motion independently until the final class score fusion step. Inspired by [8], our idea is to consider them jointly by encoding pose motion in a video clip by the directed trajectories of human joints. This results in a novel motion representation that can be directly exploited for the action recognition. Furthermore, we explore the complementary effects of depth information and thus present a 3D pose motion representation. We propose two ways to include depth information, one is multi-view 2D pose motion representation and another is 3D pose motion representation. Both representations can be directly fed to a shallow convolutional neural network to classify actions.

We evaluate our proposed method on Penn Action dataset [34]. Our experimental evaluation shows that the proposed representation outperforms other state-of-the-art pose representations. The proposed multi-view 2D PoTion outperforms original 2D PoTion while 3D PoTion underperforms 2D PoTion. We attribute this to the simplicity of our 3D CNN architecture.

1. Introduction

Action recognition is an open challenge in computer vision and has been heavily studied over the past decade. Most state-of-the-art approaches exploit the powerful Convolutional Neural Networks (CNNs) in different manners [5, 30]. A successful example is the two-stream architecture proposed by Simonyan and Zisserman [26]. Such an architecture trains two independent networks, one is for the spatial stream composed of single frame, the other is for the temporal stream composed of multi-frame optical flow. The digits from two streams are fused at the final step. As other modifications or variations are possible, the basic idea is to execute spatio-temporal convolutions on the whole scene.

In this project, however, we aim to dig the potential of human pose. Human pose has been proved to act as an important cue for action recognition [7, 16]. It can be considered as a high-level extraction of appearance that can eliminate effect of background and represent the essence of an action. Unlike traditional optical flow where movements of all pixels are given the same importance, we only focus on key human joints, which should be more reasonable and efficient. Therefore we exploit human poses for the action recognition in the project.

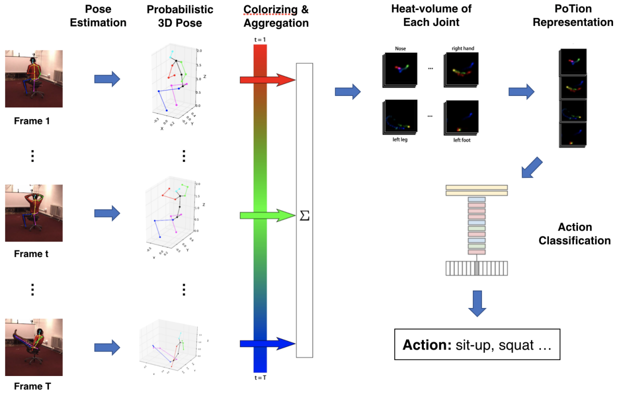

Figure 1. **Illustration of action recognition based on PoTion representation.** Given a set of frames, we extract human joints heat-volumes for each frame and colorize them to encode time information. For each joint, we aggregate them to obtain the clip-level PoTion representation and feed it to a 3D CNN model to classify actions.

While 2D pose features are helpful in action recognition [3, 24, 9], they lack depth information that could be crucial in particular cases, especially for fine-grained actions. Thus exploiting a 3D pose feature should be a more natural choice. There has been quite a few works concentrate on 3D human pose estimation [6, 2]. Most of them consists of 2D pose localization and 3D pose recovery, either processed sequentially or interactively. After testing many approaches for 3D human pose estimation, we apply the work from Zhou *et al.* [35], which is capable of performing descent 3D pose estimations on images in the wild. Since each estimated human pose is merely a static vectors containing locations of human joints in a specific frame, it is essential to combine them sequentially into a compact representation. Inspired by the work from Choutas *et al*. [8], which temporally aggregate 2D pose heatmaps into a fixed-size so-called ”PoTion” representation, we apply this idea to our 3D pose motions analogously, which results in a 3D PoTion representation for each video clip. Given this representation, a shallow CNN model can be utilized to perform action classification.

In summary, we make the following contributions:

- Extend 2D PoTion to 3D PoTion clip-level representation by temporally aggregating 3D human pose heat-volumes extracted from each frame.
- Exploit the novel 3D PoTion and multi-view 2D PoTion representations in a complete action recognition pipeline starting from video clips.

2. Related work

Action recognition has been studied considerably in the past, and it is beyond the scope of this report to provide a complete overview of the literature. In this section, we focus on relevant previous works on 3D pose estimation, motion representation and 3D CNNs.

3D Pose estimation. The 3D human pose estimator is in charge of predicting the 3D coordinates of human joints in a single image. A popular direction is to train a neural network to directly regress joint locations. Zhou *et al*. [36] explicitly enforced the bone-length constraints in the prediction, using a generative forward-kinematic layer. Tekin *et al*. [29] proposed an over-complete auto-encoder to learn a structured high-dimensional latent pose representation. Although these approaches have achieved performance gain on standard 3D pose estimation benchmark datasets, they do not generalize well to images in the wild. This is because the scene background in the wild is complicated and chaotic, images-in-the-lab-based human pose estimators tend to overfit to the simple scene background of the laboratory environment.

To solve the above outlined problem, many approaches follow a two step structure. The first step is simply estimation of 2D joint locations. The second step is the regression of the 3D locations given these 2D joints. The main advantage of this pipeline is that the 3D pose estimation network can be trained on any benchmark datasets and then adapted in other settings. For example, Pavlakos *et al*. [25] used 3D pose data and its 2D projection to train a heatmap-to-3D volumetric pose network without the original image. Chen *et al*. [6] leveraged nearest-neighbor search to match the estimated 2D pose to a 3D pose as well as a camera-view which is expected to produce a similar 2D projection from a large 3D pose library. Kocabas *et al*. [19] used their propose EpipolarPose to generate 2D poses from multi-view images, and then, utilizes epipolar geometry to obtain a 3D pose and camera geometry which are subsequently used to train a 3D pose estimator.

Motion representation. Besides the standard optical flow as input of two-stream networks, other motion representations for CNNs have also been developed. Wang *et al*. [32] propose a variant that applies warped optical flow to account for pixel motion. They also consider the difference between RGB frames as an alternative for input in order to avoid expensive optical flow computation. Yet this still limits to capture only short-term motion. Some recent approaches aim at capturing long-term motion dynamics. For instance, Sun *et al*. [28] extend LSTM by learning independent hidden state transitions of memory cells for individual spatial locations. This approach enhances the ability to model dynamics across time and introduce a novel multi-modal end-to-end system for the training procedure. Another approach, which is similar to the PoTion representation, proposed by Bilen *et al*. , introduces the concept of dynamic image, which is also a clip-level representation for action recognition [1]. This so-called dynamic image is obtained by encoding the evolution of each individual pixel across time using a rank pooling approach.

3D CNNs. In general there are two types of 3D convolutional neural networks. One is used in voxel representation of point clouds [13, 21]. The original point clouds are voxelized into binary voxel representation with fixed dimension. 3D CNN models directly operate on them to perform object recognition or semantic segmentation tasks. Due to the sparsity of point clouds, the voxels are normally of small dimension and thus the network is also shallow. Another is used in action recognition from videos [14, 33]. These methods takes the temporal information as the third dimension. They are termed as spatio-temporal convolution. [14] exploits the architectures of very deep 3D CNNs on current video datasets. They use 2D CNNs trained on ImageNet and find this produced signicant progress in various tasks.

3. Problem statement

Let X be a set of sampled frames of a video clip. Let Y be the semantic label of the given video clip. We define the action recognition problem as predicting Y given X. Note that under this formulation, the size of the frames set X may differ from different video clips, Y is unique for a given video clip and we have finite different values of Y for action recognition task on a specific benchmark.

We tackle this problem following three steps. For a given frames set X, we firstly run a state of the art 3D pose estimator [35] and then extract heat-volumes for human joints in each frame. Then we temporally aggregate and colorize them to obtain a fixed-size 3D PoTion representation for each video clip. We train a shallow 3D convolutional neural network to predict the semantic label Y.

4. Method

In this section, we present the methodology of our project in steps. We describe the human pose estimation in Section 4.1. We present the procedure to obtain 3D PoTion representation in details in Section 4.2. We present our 3D CNN architecture in Section 4.3.

4.1. Human pose estimation of images in the wild

The lack of training data imposes a great challenge on this task, as existing datasets are either images in the wild with 2D pose annotations or images in the lab with 3D pose annotations. Zhou *et al*. [35] proposed a novel end-to-end architecture to tackle the task of 3D human pose estimation in the wild. They augment the state of the art hourglass network architecture [24] by adding a 3D depth regression subnetwork. The network architecture is in Figure 2. Instead of merely feeding the output of the 2D module as input to the 3D module, they connect the 3D module with the intermediate layers of the 2D module. This allows them to share the common representations between the 2D and the 3D tasks. The network is trained end-to-end with both 2D and 3D data simultaneously. For fully-annotated 3D dataset, the training loss is the standard Euclidean loss using ground-truth depth label. For weakly-labeled 2D dataset, they propose a novel geometric constraint loss induced from the fact that ratios between bone lengths remain relative xed in a human skeleton.

They train the model on MPII dataset and Human 3.6M dataset. MPII Human Pose dataset is a large scale in-the-wild human pose dataset with 410 human activities. Human 3.6M dataset is annotated with accurate 3D joint positions captured MoCap system. We directly use the pre-trained model on Penn Action dataset without fine-tuning the parameters. For a single RGB image as input, this model outputs the 3D human joints coordinates. The 2D coordinates are given by image pixel coordinates while the depths are in metric coordinates, e.g. millimeters in this work. This model requires the human being to be centered in the input image. We crop each frame based on the given bounding box before estimating the 3D human joints coordinates. Different frames sampled from the same video clip have different bounding boxes and thus the estimated 3D coordinates are not in the same coordinate system. We transform them back to the same coordinate system as the PoTion representation is to encode the directed trajectory of human joints across different frames.

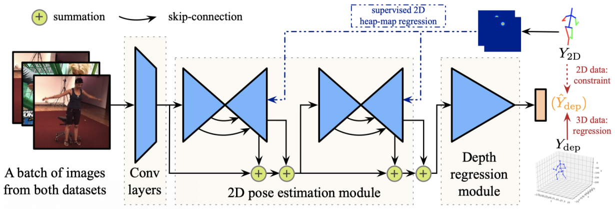

Figure 2. **Illustration of the 3D pose estimation network, adopted from [35].** Given an image, it goes through the stacked hourglass network and turn into 2D heat-maps. The 2D heat-maps together with lower-layer images features are summed as the input of the following depth regression module.

4.2. 3D PoTion representation

Choutas *et al*. proposed a clip-level pose motion representation named PoTion that considers body joints along with the image stream [18, 8]. This method produced fixed-size representation for an entire video clip, enabling manual dimensional reduction, thus suitable for later feature extraction and classification [22]. We extended this idea to 3D pose motion representation. Figure 3 gives an overview of building the 3D PoTion representation.

In the following sections, we describe the PoTion encoding method in detail, paying particular attention to extending the idea to condense 3D information. The colorization step is described in Section 4.2.1, followed by a discussion of different aggregation schemes to obtain the fixed-size clip-level representation in Section 4.2.2.

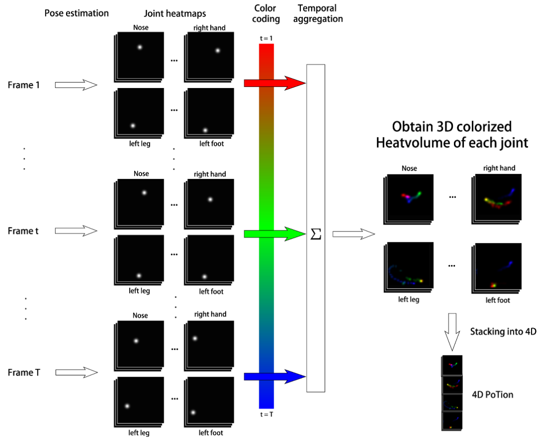

Figure 3. **Illustration of the 3D PoTion representation, adopted from [8].** Given a video, we obtain 3D joint heat volumes for each frame and colorize them based on the time information. For each joint, we aggregate them to obtain the video-level PoTion representation with fixed dimension.

4.2.1 Time-dependent heat volume colorization

After extracting the 3D joint heat volumes in each frame, we colorize them according to the relative time of this frame in the video clip. More precisely, each heat volume H_j^t of dimension $H \times W \times D$ is transformed into a volume C_j^t of dimension $H \times W \times D \times C$, i.e., with the same spatial resolution but C channels.

For $C = 2$, we can regard the channel 1 and 2 as red and green, respectively, as shown in Figure 2 (left). The idea is to encode the first frame as red, the last as green and the middle one with an equal proportion of green and red. Other frames are colorized with a linear function of the relative time t. Therefore, we get the colorize formulation $\mathrm{o(t)} = \left(\frac{t-1}{T-1}, 1 - \frac{t-1}{T-1}\right)$ for $C = 2$ channels. While we only explain the colorization scheme for $C = 2$, this approach can be easily extended to any number of color channel C. By splitting the T frames into $C - 1$ regularly sampled intervals, and in each interval, we only consider the adjacent 2 frames, as shown in Figure 4 (right) for $C = 3$.

Thus, the colorized heat volume of joint j for a pixel (x, y, z) and a channel c at time t is given by:

$$C_j^t[x, y, z, c] = \mathcal{H}_j^t[x, y, z]o_c(t) \tag{1}$$

with $o_c(t)$ being the $c - th$ element of $o(t)$.

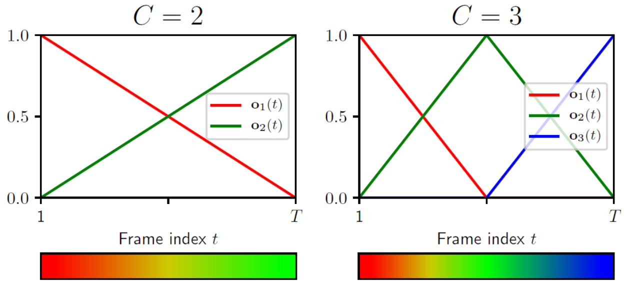

Figure 4. **Illustration of the colorization scheme for C = 2 (left) and C = 3 (right).** Figure from [8].

4.2.2 Aggregation of colorized heat volumes

We aim to encode the entire video into a dense fixed sized PoTion representation, which does not depend on the duration of the original video. Here, three different ways of aggregating the colorized heat volumes are presented. We first compute the sum of the colorized heat volumes over time for each joint j, thus obtaining a C-channel volume S_j. As S_j is an aggregation of various frames that depends on the frame number T, to obtain an invariant representation, each channel c is normalized by dividing by the maximum value over all pixels. The PoTion representation is thus obtained:

$$\mathcal{P}_j[x, y, z, c] = \frac{S_j[x, y, z, c]}{\max S[x, y, z, c]} \tag{2}$$

With the PoTion representation, if a joint stays at a given position for some time, a stronger intensity will be accumulated, which could be detrimental for further feature extraction. Thus, a second variant with normalized intensity is proposed. The intensity I_j is calculated by summing the values of all C channels for every pixel:

$$I_j[x, y, z] = \sum_{c=1}^{C} \mathcal{P}_j[x, y, z, c] \tag{3}$$

A normalized Potion representation can then be obtained by dividing P_j by the intensity I_j.

$$\mathcal{N}_j[x, y, z, c] = \frac{\mathcal{P}_j[x, y, z, c]}{I_j[x, y, z]} \tag{4}$$

The original paper reported that stacking all three representations increases the video classification accuracy by 0.3% on HMDB-1 dataset [17] but decreases by 1.8% on JHMDB-1 dataset [16]. We use PoTion and normalized PoTion representations in our experiments.

4.3. Shallow 3D CNN model

We only apply shallow 3D CNN architectures in our task as the 3D PoTion representation has significantly less texture than standard RGB images and its frame size is much smaller (32 in our case). We don't use any pre-train model on ImageNet [10] and train from scratch as there is a domain shift from normal RGB images to our proposed PoTion representation.

In our task, the PoTion representation is more similar to binary voxels studied in point clouds interpretation [21]. Therefore, we adopt the 3D CNN architecture proposed in [13]. The input to the network has $M(C + 1)$ channels. M is the number of human joints. The structure of this shallow 3D CNN is visualized in Figure 5.

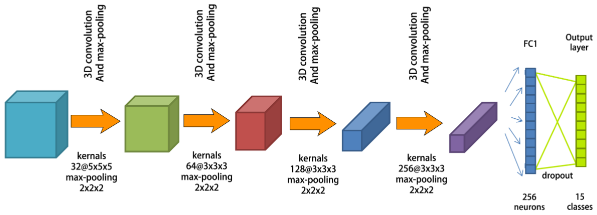
3D convolution And max-pooling
kernals 32@5x5x5 max-pooling 2x2x2
3D convolution And max-pooling
kernals 64@3x3x3 max-pooling 2x2x2
3D convolution And max-pooling
kernals 128@3x3x3 max-pooling 2x2x2
3D convolution And max-pooling
kernals 256@3x3x3 max-pooling 2x2x2
FC1
Output layer
dropout
256 neurons
15 classes

Figure 5. **Shallow 3D convolutional neural network**

5. Experimental results

In this section, we present the action recognition results on Penn Action dataset. We first present the 3D pose estimation results in Section 5.1. Then we present the obtained PoTion representations in Section 5.2. Finally we present the action recognition results in Section 5.3.

5.1. Pose estimation

We estimate the 3D human poses of all the frames in Penn Action dataset. The method proposed in [35] was open-sourced and implemented with torch 7. As only 2D annotations are provided, We only compare the 2D pose estimations. The quantitative results are in Table 1 and the qualitative results are in Figure 6.

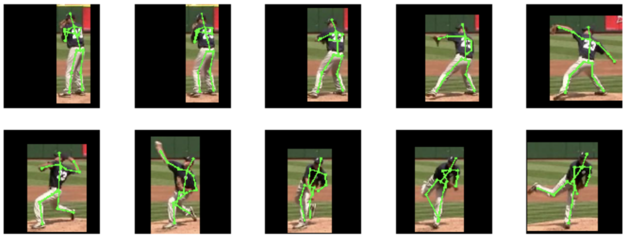

Figure 6. **Pose estimation results in 2D.** The green lines are the estimated skeleton. We crop each frame to center the human being and the black background represents the global bounding box.

5.2. PoTion representation

With estimated 3D human joint coordinates, we obtain its 3D heat-volumes by applying a Gaussian kernel centered at each joint to simulate the probability distribution for the joint being at a specific location. We set the standard deviation to be 1.5 and the amplitudes to be a random number between 0.33 and 0.37. We experiment with different frame sizes. As visualizing 3D voxels is difficult in 2D, we only present 2D PoTion here. We can observe from Figure 5.2 that the frame size of 32 tends to yield blurred representation of that from frame size of 96.

5.3. Action recognition

Datasets. Penn Action dataset [34] is composed of 2326 videos in the wild with 15 different actions, among those ”bowling”, ”gold swing”, ”squats”, etc. The challenge on this dataset is that several body parts are missing in many actions and the image scales vary from one sample to another. 2D human joint coordinates, action labels as well as bounding boxes are provided. We use 881 samples for training, 377 samples for validation and 1068 samples for test.

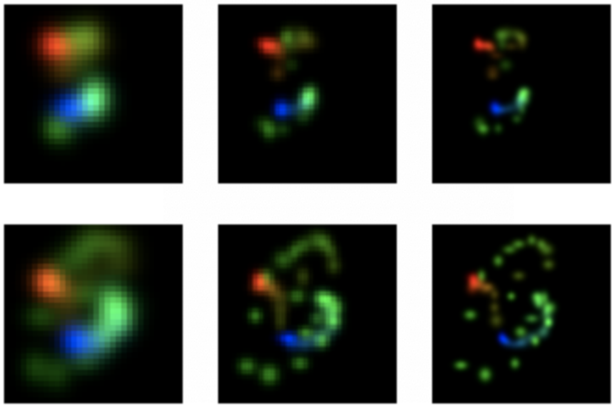

Figure 7. **PoTion representations of different frame sizes.** Left to right: 32, 64, 96.

Network architectures. In the case of 2D CNN, We define a convolutional block $CB(m)$ as follows: $C(m,3,2)-BN-C(m,3,1)-BN$, where $C(m,l,n)$ denotes a convolutional layer with n strides and m $l \times l$ filters, BN denotes batch normalization layer. Each batch normalization layer is activated by ReLU activation function [23]. When frame size is 32, the architecture is as follows: $CB(32)-GP-FC(15)$, where GP denotes global max pooling layer, $FC(m)$ denotes fully connected layer with m neurons. Different from the original paper, we remove the dropout layer [27] in convolutional blocks as we observe a decrease in accuracy by 2% when we drop with a probability of 25 %.

In the case of 3D CNN, the architecture is as follows: $C(32,3,1)-P(2)-C(64,3,1)-P(2)-c(128,3,1)-P(2)-FC(512)-FC(15)$, where $P(m)$ denotes a max pooling layer with m strides. Each convolutional layer is followed by a dropout layer with dropout rate of 25%.

Implementation details. We initialize all layer weights with Xavier initialization [12]. We optimize the network with Stochastic Gradient Descent (SGD) with nesterove momentum. The batch size is 32. We train our neural network model on a NVIDIA Titan X GPU with 12GB RAMs.

In the case of 2D CNN, we initialize the learning rate with 0.05, the momentum is 0.9 and the decay rate is 0.09. We train for 100 epochs and save the model with the smallest validation loss.

In the case of 3D CNN, we initialize the learning rate with 0.003. We scale it by a factor of 0.6 with 6 patience. We early stop the training with 10 patience.

Impact of pose estimation and frame size. We compare PoTion representations extracted from estimations and ground truth. The final results are in Table 2. We ob-

	BaseballPitch	CleanAndJerk	PullUp	StummingGuitar	BaseballSwing	GoldSwing	PushUp	Situp
accuracy	15	21	35	135	11	11	79	76
	BenchPress	JumpingJacks	TennisForeHand	TennisServer	Bowl	JumpRope	Squat	
accuracy	97	17	11	12	20	38	79	

Table 1. **2D pose estimations results** The estimations are compared against the ground truth, the metric is mean Euclidean loss in pixels grouped by different actions.

Frame size	# Params	PoTion	Accuracy
32	0.08M	pred	94.64
		gt	**95.64**
64	0.29M	pred	94.95
		gt	95.30
96	1.17M	pred	95.12
		gt	95.13

Table 2. **Action recognition results using 2D PoTion representation.** pred indicates that PoTion is extracted from estimations, gt indicates that PoTion is synthesized from ground truth annotation.

Method	#Params	Voxel	Accuracy
VoxNet[21]	2.08M	binary	85.51
		continuous	**87.12**
Hackel [13]	1.06M	binary	91.07
		continuous	**91.57**

Table 3. **Action recognition results on 3D PoTion representation.**

Flip	2D-gt-32	2D-pred-32	2D-multiview	3D-pred-32
yes	95.56	**94.78**	95.01	91.56
no	**95.64**	94.64	**95.51**	**91.57**

Table 4. **Action recognition results with and without data augmentation**

Inputs	Methods	Accuracy
motion	Iqbal *et al.* [15]	79.0
	Walker *et al.* [31]	21.2
	Gao *et al.* [11]	52.4
	2D PoTion[8]	95.12
	Multi-view 2D PoTion(Ours)	**95.51**
	3D PoTion(Ours)	91.57
RGB+motion	Cao *et al.* [4]	95.3
	Luvizon *et al.* [20]	**97.4**

Table 5. **Comparison against the state-of-the-art results on Penn Action dataset.**

serve that different frame sizes almost have the same performance on this task, while smaller frame size requires much fewer parameters. We also observe that there is around 1% gain in accuracy using PoTion representations synthesized from ground truth annotations compared to estimations when frame size is 32. The difference is eliminated when increasing the frame size.

Impact of 3D CNN architectures. There are two choices of 3D CNN architectures [13, 21] and we exploit both. Our 3D PoTion representation contains continuous values, this is contrary to binary voxels used in [13]. We binarize them with a threshold of 0.5 and the complete comparison is in Table 3. We observe that [13] outperforms [21] while requires only half of the parameters. Binarizing the PoTion representation yields a drop in both models, this is reasonable as this operation actually loses importance information.

Impact of data augmentation. We present the influence of data augmentation during training in Table 4. We randomly horizontally flip and switch the pairs of left and right human joints to augment the PoTion representations. We observe that this data augmentation strategy is not effective in this dataset. We attribute this to the simplicity of this dataset that left and right parts play small role in distinguishing between different actions.

Comparison against SOTA results. We compare our proposed 3D PoTion and multi-view 2D PoTion with the state of the art. We observe in Table 5 that our proposed multi-view 2D PoTion achieves the best performance using motion representation only. Luvizon *et al.* [20] outperforms ours by around 2% in accuracy by combining appearance stream and motion stream. Exploiting appearance stream is beyond the scope of our project, therefore we discuss motion stream only. However, it's still worthy of noticing that the original paper reported that combining PoTion and I3D[5] could yield a gain of 0.3 % in accuracy on [17].

6. Conclusion

This paper extends the 2D PoTion originally proposed in [8] to 3D PoTion by adding the depth information. We explore two ways to introduce it which are 3D PoTion representation and multi-view 2D PoTion representation. We show that multi-view 2D PoTion achieves the state of the art results on Penn Action dataset. Future work includes exploiting the complementary effect of multi-view 2D PoTion to appearance stream in action recognition and exploring better 3D CNN architectures.

References

[1] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016. 2

[2] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 1

[3] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016. 1

[4] C. Cao, Y. Zhang, C. Zhang, and H. Lu. Body joint guided 3-d deep convolutional descriptors for action recognition. *IEEE transactions on cybernetics*, 48(3):1095–1108, 2017. 6

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 6

[6] C.-H. Chen and D. Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017. 1, 2

[7] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3218–3226, 2015. 1

[8] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7024–7033, 2018. 1, 2, 3, 4, 6

[9] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017. 1

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 4

[11] R. Gao, B. Xiong, and K. Grauman. Im2flow: Motion hallucination from static images for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5937–5947, 2018. 6

[12] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 5

[13] T. Hackel. *Large-scale Machine Learning for Point Cloud Processing*. PhD thesis, ETH Zurich, 2018. 2, 4, 6

[14] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 2

[15] U. Iqbal, M. Garbade, and J. Gall. Pose for action-action for pose. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 438–445. IEEE, 2017. 6

[16] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013. 1, 4

[17] H. Jhuang, H. Garrote, E. Poggio, T. Serre, and T. Hmdb. A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, volume 4, page 6, 2011. 4, 6

[18] H. R. V. Joze and O. Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053*, 2018. 3

[19] M. Kocabas, S. Karagoz, and E. Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[20] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018. 6

[21] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 2, 4, 6

[22] W. McNally, A. Wong, and J. McPhee. Star-net: Action recognition using spatio-temporal activation reprojection. *arXiv preprint arXiv:1902.10024*, 2019. 3

[23] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 5

[24] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 1, 3

[25] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017. 2

[26] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 5

[28] L. Sun, K. Jia, K. Chen, D.-Y. Yeung, B. E. Shi, and S. Savarese. Lattice long short-term memory for human action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2147–2156, 2017. 2

[29] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016. 2

[30] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017. 1

[31] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2443–2451, 2015. 6

[32] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[33] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*, 2015. 2

[34] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2248–2255, 2013. 1, 5

[35] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017. 1, 2, 3, 5

[36] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. In *European Conference on Computer Vision*, pages 186–201. Springer, 2016. 2