PAS-Net: Pose-based and Appearance-based Spatiotemporal Networks Fusion for Action Recognition

Changzhen Li, Jie Zhang, Shiguang Shan, Xilin Chen
Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
University of Chinese Academy of Sciences, Beijing 100049, China

***Abstract*— Human poses play important roles in action analysis. However, most state-of-the-art approaches in action recognition ignore the importance of human poses and rarely leverage the pose information for further improving the recognition performance. In this paper, we propose a novel network architecture, which simultaneously considers the appearance information and pose knowledge for robust action recognition. We explore various architectures for fusing the appearance and pose information rather than simply averaging scores at the final layer. Moreover, a novel training strategy is proposed to reduce the influence of overfitting for limited training data. Extensive experiments show that our method achieves competitive performance on the popular benchmarks, *i.e.*, UCF-101 and HMDB-51.**

I. INTRODUCTION

Both human action analysis and pose estimation have been receiving considerably more attention in computer vision recently, on account of their wide and practical applications in daily life, such as intelligent surveillance and human-computer interaction. And human poses convey crucial messages in action recognition, which helps to eliminate the influence by clutters or non-human motions from backgrounds and to handle appearance variations from scale transformations, viewpoint changes, camera motions, *etc*. [14], [39]. It has been shown that poses indeed provide complementary information to appearance and motion for action analysis [4], [17].

Action recognition benefits a lot from great progresses in deep learning. In particular, the Two-stream ConvNets and 3D ConvNets architectures improve the performance in the video classification task greatly [24], [26]. In spite of the great potential of poses, most state-of-the-art approaches [3], [28], [33] are built upon 3D ConvNets by utilizing appearance information and usually also optical flows. However, the optical flows typically need to be computed ahead of time, which prevents assembling an end-to-end learning scheme. Moreover the entire 3D CNNs bring heavy computation and memory burden, *e.g.*, Tran et al. [26] spend two months on training them. There exist few pose-based approaches [5], [37] using pose information as input modality for action recognition. And several works convert pose features into skeleton keypoints and model Graph Convolutional Network(GCN) on skeleton sequences for classification [38], which completely neglects the relationship between persons and objects or backgrounds. Several pose-based approaches take pose information into account [20], [37], which simply average scores at the softmax layer and few works [9] are devoted to exploring different architectures for fusing different input modalities.

In this paper, to make full use of pose information and handle the above-mentioned issues, we propose a novel pose-based and appearance-based spatiotemporal convolutional network architecture (PAS-Net), which explores various fusing methods to integrate pose and appearance information for action recognition. We attempt to disentangle appearance features (such as texture and color) and pose features (such as shape and body articulation) before modeling temporal features, which can simplify the complexity of classification tasks caused by the superposition of different variable factors in natural images [21]. Specifically, the proposed architecture consists of four subnetworks as depicted in Fig. 1 : Appearance ConvNet, Pose ConvNet, Fusing Network and Spatiotemporal ConvNet. Appearance 2D ConvNet extracts abundant multi-scale appearance features through effective Inception architecture [15]. Pose 2D ConvNet acquires ample pose information, including implicit pose feature, explicit pose heatmap (keypoints) and part-affinity-fields (PAF) feature by introducing fast and efficient Openpose [2]. Then appearance and pose features are aggregated in Fusing Network. Finally, Spatiotemporal 3D ConvNet leverages the integrated appearance and pose information to learn spatiotemporal features for action recognition. The pose information can be further improved to be more favorable for action recognition under an end-to-end training manner. Our PAS-Net does not need to compute optical flow [1], trajectories [29] or other auxiliary information ahead of time like those in Two-stream ConvNets. Furthermore, although another pose stream is appended to our PAS-Net, the hybrid 2D/3D Convnet is computationally efficient at a low time and memory cost. So our PAS-Net is an effective and efficient end-to-end trainable architecture for action recognition by only taking raw video as input.

Additionally, our network adopts a sparse temporal sampling strategy [32], based on the observations that consecutive frames are highly redundant and the action of video sequences can be straightway concluded from a single frame. And this strategy helps to model long-range temporal features at a low time and memory cost. Then, appearance and pose features from each sampled frame are computed

This work was partially supported by Natural Science Foundation of China under contracts Nos. 61806188.

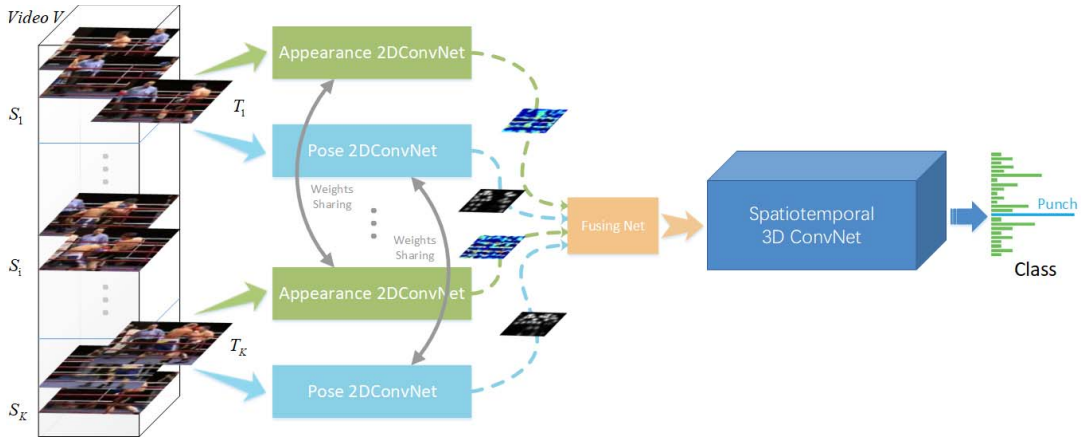


Fig. 1. **PAS-Net** is an effective and efficient end-to-end trainable architecture. Appearance ConvNet extracts abundant multi-scale appearance features by Inception architecture. Pose 2D ConvNet acquires ample pose information by fast and efficient Openpose. Then appearance and pose features are aggregated in Fusing Network. Spatiotemporal 3D ConvNet leverages the integrated appearance and pose information to learn spatiotemporal features for action recognition.

via feature extraction network, but one thing to note here is that their temporal features still retain in the correlation among sampled frames along the temporal dimension. So an essential difference between simple mixed 2D/3D Convnet [28] and our PAS-Net is that they process the whole stacked frames as input rather than single frame like ours, which collapses the temporal information of the video in single-channel feature maps due to the limitation of 2D CNNs. Moreover, we attempt to align the responses of appearance and pose features at the same pixel position in Fusing Network, and study how these evolve along time with different input modalities and fusing architectures in Spatiotemporal ConvNet. Complicated networks [33], [42] easily bring the risk of severe overfitting to small datasets if training from scratch, which is usually solved by pretraining on large datasets firstly and then finetuning on the small target datasets. Differently, we present a twice-finetune training strategy to settle out the overfitting problem.

We conduct comprehensive experiments on the popular benchmarks, *i.e.*, UCF-101 [25] and HMDB-51 [19]. Benefited from leveraging pose information, our network demonstrates a significant improvement, which implies the effectiveness of our architecture. Although no auxiliary information like optical flow is used, we achieve competitive performance compared to the state-of-the-art approaches.

II. RELATED WORK

A. Action Recognition

Traditional hand-crafted representation approaches for action analysis, such as iDT [30], are computationally expensive and inefficient on extracting context and high-level features. Afterward, Convolutional Neural Networks have exhibited excellent performances in video analysis, including action recognition, action detection and video captioning. Recently there are two typical methods for action recognition: a) Two-stream ConvNets [24], b) 3D ConvNets [26].

Two-stream ConvNets is first proposed by Simonyan et al. [24], which processes RGB inputs and optical flows in different branches separately. Feichtenhofer et al. [9] propose different fusing architectures to take advantage of both RGB inputs and optical flows information. Temporal Segment Networks presented by Wang et al. [32] use a sparse temporal sampling strategy to capture long-term temporal representations. Feichtenhofer et al. [8] introduce residual connections to allow spatiotemporal interaction between the two streams. Although Two-stream ConvNets achieves promising results for action recognition, the optical flows are hand-crafted representations which are computationally expensive to obtain. Moreover, the optical flows are always computed ahead of time, which restricts end-to-end trainable abilities to some extent. Compared to the above methods, our PAS-Net captures pose information by Pose ConvNet and appearance features by Appearance ConvNet on raw videos, which is a fully end-to-end architecture for robust action recognition.

The other influential method, 3D ConvNets, has shown favorable capabilities for action recognition [26]. Deeper backbones, such as DenseNet [13] and ResNet [12], are considered in 3D CNNs as T3D [6] and R3D [27] respectively, which demonstrates the superiority of 3D CNNs over 2D CNNs. Two-Stream Inflated 3D ConvNet (I3D) presented by Carreira et al. [3] leverages the advantages of both two-stream inputs and 3D ConvNet, which shows considerably impressive performance. Zolfaghari et al. [42] capture appearance features in 2D CNN and acquire temporal context in 3D CNN for recognition at an 80x faster rate than I3D [3]. The main drawback of 3D ConvNets is oversize parameters and memory cost, making it easy to overfit on small datasets that are yet common situations in practical applications. One effective solution is to explicitly factorize 3D convolution into 2D spatial convolution and 1D temporal convolution [23], [28]. Compared to these entire 3D ConvNets, our PAS-Net, a two-stream hybrid 2D/3D Convnet, not only alleviates oversize parameters and memory cost but also extracts spatiotemporal features more efficiently for the disentanglement of appearance and pose.

Besides, another interesting approach in action recognition [35] models the temporal relation among frames by means of

recurrent neural network(RNNs). And other input modalities like warped flow [32], RGB difference [32], object information [16], and audio [36] are utilized as complementary information to improve classification performance. [10] endeavors to explore regions of the person of interest and their context by action transformer architecture to recognize actions. [31], [33], [34] attempt to model more effective spatiotemporal relations for action recognition.

B. Pose-related Action Recognition

Human poses, captured by current methods, are sufficient to provide the discriminative clues for action recognition. Recently, there appear some attempts for pose-related action recognition. Choutas et al. [5] introduce a novel representation that encodes the motion of semantic keypoints. Yan et al. [38] compute skeleton joint coordinates and construct ST-GCN on skeleton sequences. Du et al. [7] model RNNs for classification utilizing pose-attention mechanism. Luvizon et al. [22] propose a multitask framework for jointly pose estimation and human action recognition. Yan et al. [37] take pose information into account for classification but integrate different streams by simply averaging, and analogous works proposed in [20]. Although methods above leverage pose information for action recognition, few specialize in how to fuse pose and appearance information more effectively. Compared to these methods, our method not only proposes a hybrid convolutional architecture but also considers various fusion methods for fusing pose and appearance information to improve the action recognition.

III. PROPOSED METHOD

Our PAS-Net is a two-stream hybrid 2D/3D ConvNet which simultaneously leverages appearance and pose information. We firstly provide a detailed introduction of our PAS-Net. Then we discuss the necessity for aligning responses of appearance and pose features, and explore various fusing methods for fusing them. Finally, we describe the training strategies and implementation details for learning PAS-Net.

A. PAS-Net

As illustrated in Fig. 1, our PAS-Net has three main stages: features extraction, features fusion, action recognition. The features extraction is further subdivided into appearance extraction and pose extraction. Our network takes the visual information of the whole video as input and provides a video-level prediction as output, which is an entire and efficient end-to-end architecture. The overall objective can be formulated as:

$$\text{PAS-Net}(V) = N_{st}\left(N_{\text{fuse}}\begin{pmatrix} N_{rgb}\left(T_1\right), N_{pose}\left(T_1\right) \\ N_{rgb}\left(T_2\right), N_{pose}\left(T_2\right) \\ \cdots \\ N_{rgb}\left(T_K\right), N_{pose}(T_K) \end{pmatrix}\right). \quad (1)$$

Firstly the whole input video V is grouped into K segments $\{S_1, S_2, ..., S_K\}$ with the same number of frames. Then an unique frame T_i is sampled randomly from each segment S_i representing this segment. Next, the appearance extraction network N_{rgb} and the pose extraction network N_{pose} capture appearance and pose features from each sampled frame. Furthermore, the appearance and pose features along time are further integrated by employing Fusing Network N_{fuse}. Spatiotemporal ConvNet N_{st} leverages the integrated features and learns spatiotemporal features representation of the whole video. Finally, softmax function is adopted to estimate the probability of each action.

1) Appearance ConvNet (N_{rgb}): Appearance ConvNet utilizes Inception architecture to capture multi-scale appearance features, whose initialized weights come from ECO-($\mathcal{H}_{2D}$) [42]. Moreover, Appearance ConvNet processes a single frame at a time, rather than stacked images in simple mixed 2D/3D Convnet [28]. So it only captures spatial features, and temporal features still retain the correlation among sequential frames in order.

2) Pose ConvNet (N_{pose}): Pose ConvNet utilizes VGG-19 architecture to capture robust pose features, the initialized weights of which come from Openpose [2]. We use the feature maps of the 6th stage as the pose information. Specifically, the implicit pose features come from 6-6 convolutional layer features in the detection branch, the explicit pose features (keypoints) come from the maximum of 6-7 convolutional layer features in channel dimension in the detection branch, and the part-affinity-fields features come from 6-6 convolutional layer features in the association branch. Besides, Pose ConvNet processes a single image at a time in a similar way as Appearance ConvNet, so the same output resolution between them can be easily constructed for succedent feature fusion.

3) Fusing Network (N_{fuse}): Fusing Network tries various fusion methods, including different combinations of inputs and operators, and the details are described in Sec. III-B.

4) Spatiotemporal ConvNet (N_{st}): In Spatiotemporal ConvNet, 3D CNNs are considered to capture spatiotemporal representations for its efficiency. Spatiotemporal ConvNet utilizes 3D Resnet architecture [27], and the initialized weights of which come from ECO-($\mathcal{H}_{3D}$) [42]. Furthermore, the last layer of Spatiotemporal ConvNet is 3D global pooling, and its output feature is a 512-d vector which is the spatiotemporal representation of the entire video. Since temporal information modeling gets processed only in Spatiotemporal ConvNet, we disentangle and extract appearance features and pose features without temporal interference from other frames.

B. Alignment and Fusion

Here we firstly take into account aligning responses of appearance information and pose knowledge, and then explore different architectures for fusing them. Furthermore, we attempt to settle out the challenges of where to fuse, what to fuse and how to fuse. Subsequent empirical experiments will evaluate the performance in different fusion architectures in Sec IV-B.

Before fusion, it is very essential and meaningful to align responses in three factors: temporal dimension, semantic dimension, and spatial dimension. And typically in practice, the video frame stands for temporal dimension,

channel capacity and input resolution can be interpreted as semantic dimension and spatial dimension respectively. So we choose to fuse appearance and pose features at the last 2D convolutional layer, because temporal information is not dealt with until entering Spatiotemporal ConvNet. In this way, appearance features can be integrated with pose features at the same time without temporal interference from other frames, and temporal relation is in correspondence strictly. Meanwhile the spatial correspondence is also easily achieved, as long as two steams have the same feature resolution, which can be simply implemented by pooling or convolution with strides. Furthermore, we believe that optimizing the trainable filters of the network can fits channel correspondence between appearance and pose features gradually to some extent. By the above careful design, we align responses of appearance and pose features in temporal, spatial and semantic dimensions.

To specify fusion methods in detail, the concepts and formal definitions related to appearance and pose features are presented. Note, only one segment T_i is considered to simplify the issue.

$$\begin{aligned} h_i &= N_{\text{fuse}}\left(N_{\text{rgb}}\left(T_{\text{i}}\right), N_{\text{pose}}\left(T_{\text{i}}\right)\right) \\ &= N_{\text{fuse}}\left(\left(x_{\text{A}}^i\right), \left(x_I^i, x_E^i, x_F^i\right)\right). \end{aligned} \tag{2}$$

Here $T_i \in R^{3\times h\times w}$ is the sampled frame. $x_A^i \in R^{c\times h\times w}$ is appearance feature computed by Appearance ConvNet. $x_I^i, x_F^i \in R^{c\times h\times w}$, $x_E^i \in R^{1\times h\times w}$ are implicit pose feature, part-affinity-fields feature and explicit pose feature (keypoints) computed by Pose ConvNet. $h_i \in R^{c\times h\times w}$ is hybrid feature computed by fusing Network, and c, h, w is the number of channels, width, height in the feature map. N_{fuse} is a complicated function representing Fusing Network, which can be simple operations like addition and multiplication, or complex networks like convolution and pooling, and it will be considered concretely based on different inputs in the following:

1) Appearance feature x_A or pose features x_I only:

$$h = x_A. \tag{3}$$

$$h = x_I. \tag{4}$$

Equation (3) only uses the appearance feature as input like ECO [42], we set it as a **baseline** for comparison. And Equation (4) only utilizes the pose feature as input.

2) Explicit pose heatmap x_E and appearance feature x_A:

$$h = x_E \otimes x_A, \tag{5}$$

$$h = x_A + \lambda_1 \cdot x_E \otimes x_A, \tag{6}$$

$$h = conv\left(cat\left(x_A, x_E \otimes x_A\right)\right). \tag{7}$$

Equation (5) computes the pixel wise product of x_E and x_A, and x_E represents pose probability map. In other words, joints probability map as attention map is attached to appearance feature. Observing the sparsity of pose heatmap, we integrate original appearance feature x_A and the product in (5) once more. Equation (6) computes the sum of x_A and the product in (5), where λ_1 is a hyper-parameters to balance them. Equation (7) concatenates these two features first, then convolves the stacked data with filter f.

3) Implicit pose feature x_I and appearance feature x_A:

$$h = x_A + \lambda_2 \cdot pool\left(x_I\right), \tag{8}$$

$$h = conv\left(cat\left(x_A, x_I\right)\right). \tag{9}$$

Equation (8) computes the sum of x_I and x_A, where mean-pooling operation is to fit the number of channels in correspondence. Equation (9) computes the result of convolution of concatenation between x_I and x_A. And others are the same as above.

4) Implicit pose feature x_I, part-affinity-fields feature x_F and appearance feature x_A:

$$h = x_A + \lambda_3 \cdot pool\left(x_I\right) + \lambda_4 \cdot pool\left(x_F\right), \tag{10}$$

$$h = conv\left(cat\left(x_A, x_I, x_F\right)\right). \tag{11}$$

Equation (10) computes the sum of x_I and x_F and x_A, and equation (11) computes the convolution of concatenation among them. And others are the same as above.

C. Training Strategy

Complicated networks, in particular 3D CNNs, are more likely to take a severe risk of overfitting on small datasets that are yet common situations in practical applications. Before large datasets like Kinetics [18] are presented, many attempts including multitask learning [24], high dropout [24], cross modality pre-training [32], regularization techniques [32], data augmentation [32] are conducted to mitigate the impact of overfitting. Currently almost all works [3], [6], [28] choose to pretrain on large datasets firstly and then finetune the model on small target datasets, so training with small datasets relies too much on initialization weights from large datasets.

Consider the scenario: The network N_0 (ECO) is a well-known network architecture, whose weights w_0 can be initialized from existing models pretrained on huge datasets D_0 (Kinetics). We design a more efficient network N_1 (PAS-Net) based on ECO, which aims to obtain good performance on the small target dataset D_1. There are three training strategies:

Conventional:
1. finetune N_1 on D_1 through $w_0 \rightarrow w$
Recent:
1. finetune(1st) N_1 on D_0 through $w_0 \rightarrow w_1$
2. finetune(2nd) N_1 on D_1 through $w_1 \rightarrow w$
Our twice-finetune:
1. finetune(1st) N_0 on D_1 through $w_0 \rightarrow w_1$
2. finetune(2nd) N_1 on D_1 through $w_1 \rightarrow w$

The conventional methods directly finetune N_1 on small target datasets, which always bring about the performance degeneration caused by overfitting; The recent methods, *i.e.*, I3D and ECO, firstly pretrain on huge datasets D_0 then train on small target datasets, which always takes quite expensive time and memory cost; Our twice-finetune firstly finetune original network N_0 on target datasets D_1 with initializations from the model pretrained on D_0, then train N_1 on target datasets D_1. It achieves excellent performance with no need for pertraining N_1 on huge datasets D_0, which demonstrates our twice-finetune is both effective and efficient.

D. Implementation Details

We train our networks using mini-batch stochastic gradient descent (SGD) with a momentum of 0.9, and weight decay of $5e^{-4}$. The initial learning rate is 0.001 and the dropout rate is 0.3. And the hyper-parameters λ_1,λ_2,λ_3,λ_4 in Fusing Network are set to 1.5, 1.5, 0.5, 1. The implementation of our PAS-Net is based on PyTorch with TITAN Xp GPU.

During training, all sampled frames are firstly resized to 256*340, and then data augmentations including random cropping and horizontal flipping are employed, finally the cropped regions are resized to 224*224 for training. When testing, only applying center cropping, we sample K frames from each video to compute the final prediction result. And we also provide an ensemble model with $\{16, 20, 24, 28, 32\}$ frames like ECO [42].

IV. EXPERIMENTS

We conduct comprehensive experiments on two popular datasets, UCF-101 [25] and HMDB-51 [19]. Firstly we make brief introductions of the datasets. Then we explore the influence of various training strategies, fusing methods, and frame numbers. Moreover we compare our method to the state of the art methods. Finally, we visualize the learned pose information in our PAS-Net to further demonstrate the effectiveness of leveraging poses for action recognition.

A. Datasets

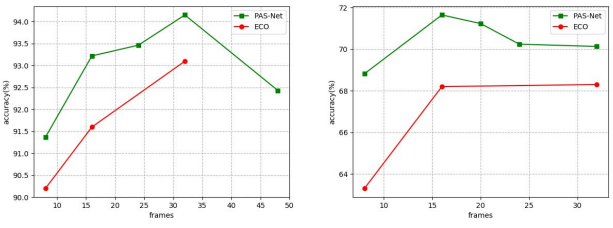
We conduct experiments on UCF-101 and HMDB-51 following most works in pose-based action recognition. **UCF-101** [25] consists of over 13k clips and 27 hours of video data from 101 action classes, which contains realistic user-uploaded videos from YouTube. **HMDB-51** [19] consists of 6766 video clips from 51 action classes from YouTube. Both of them involve daily activities containing variations from camera motion, viewpoint, and occlusion.

B. Ablation Studies

1) Training strategies: Table I shows two cases of training strategies, the conventional and twice-finetune, with 3 most typical fusion instances on UCF-101, *i.e.*, appearance features(baseline), the convolution based fusion of appearance and pose features, the convolution based fusion of appearance, pose and PAF features. Compared to the baseline, the performance degenerates severely when using the conventional finetuning strategy. In particular, we add a simple 1*1 convolutional layer to ECO as a controlled trial. Although the modification of network architecture is slight, the performance also degrades, which demonstrates that a slight alteration of the original network may lead to poor performance as described in Sec. III-C. In contrast to the conventional finetuning strategy, the proposed twice-finetune training strategy significantly improves the performance for all cases and slightly surpasses the baseline. An interesting discovery is that the more complex the network is, the higher accuracy is achieved with the twice-finetune strategy and meanwhile the lower accuracy by the conventional finetuning strategy. The possible reason behind it is a more complex network with pose information indeed helps to deal with the action recognition, but it is quite difficult for the conventional finetuning strategy to approach the optimal solution due to overfitting.

2) Fusion methods: We conduct experiments to evaluate nine different fusion methods as described in Sec. III-B, and Table II show the results on UCF-101. As can be seen, only utilizing pose information (I) leads to the severe performance degeneration. And the performance of calculating the pixel-wise product of appearance and pose probability map ($\mathrm{E} \circ \mathrm{A}$) have a similar decline for the sparsity of pose probability map. Moreover, When the product ($\mathrm{E} \circ \mathrm{A}$) is appended to original appearance features, we achieve an obvious performance improvement up to 90.94% and 90.81% than the baseline for the addition based and convolution based fusions respectively, which demonstrates the advantages of fusing body pose information. Simple addition operation obtains better performance than the convolution, it is probably because that to learn the parameters of convolutional kernels is rather difficult in consideration of the limitation from small datasets. Moreover, the utilization of implicit pose feature, part-affinity-fields feature and appearance feature($\mathrm{A} + 0.5 \cdot \mathrm{pool(I)} + \mathrm{pool(F)}$) achieves the best results and all the following experiments utilize this fusion style.

3) Frame numbers: Extensive experiments are conducted to investigate the influence of various frame numbers K, *i.e.*, 8, 16, 24, 32 and 48. As depicts in Fig. 2, we concluded that the classification accuracy is improved along with the increase of frame numbers, since more information of the entire video is leveraged. Our PAS-Net with 16 frames achieves the best classification accuracy on HMDB-51 of 71.6% while it with 32 frames achieves the best result of 94.1% on UCF-101. The cause for differences might be that different datasets have different video length, and we discover that the average length of video in UCF-101 is 186.36 frames while that in HMDB-51 is 95.16 frames. When continuing to increase frame numbers, the accuracy decreases. One possible reason for this decrease is that more frames contain more redundant information, which may be harmful for model learning.



(a) (b)

Fig. 2. Performance comparison of various frames on UCF101(left) and HMDB51(rights)

C. Comparison with the State-of-the-art Results

We compare our PAS-Net with the state-of-the-art methods on both UCF101 and HMDB51. Tabel. III summarizes the accuracy comparisons results when using only RGB modality. Our PAS-Net outperforms TSN [32], DTPP [40], ARTNet [31] and ECO [42] on both UCF101 and HMDB51

TABLE I

PERFORMANCE COMPARISON BETWEEN THE CONVENTIONAL FINETUNING STRATEGY AND OUR TWICE-FINETUNE TRAINING STRATEGY ON UCF101.

Training strategies \ Fusion Style Acc(%)	A (baseline)	A + 1*1conv	Conv(A, I)	Conv(A, I, F)
Conventional	90.20	78.73	77.54	77.22
Twice-finetune	-	90.28	90.44	**90.89**

TABLE II

PERFORMANCE COMPARISON OF DIFFERENT FUSION METHODS (SEC. III-B) ON UCF101

Fusion Style	Accuracy(%) top-1	top-5
A (baseline)	90.20	-
I	57.44	83.82
E ∘ A	78.10	94.65
A + 1.5·E ∘ A	90.94	98.17
Conv(A, E ∘ A)	90.81	98.17
A +1.5· pool(I)	91.15	98.20
Conv(A, I)	90.44	98.23
A + 0.5·pool(I) + pool(F)	**91.37**	98.04
Conv(A, I, F)	90.89	98.22

TABLE III

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON UCF101 AND HMDB51 USING ONLY RGB MODALITY.

Method	Pre-training models	Accuracy(%) UCF101	HMDB51
I3D [3]	ImageNet	84.5	49.8
TSN [32]	ImageNet	86.4	53.7
DTPP [40]	ImageNet	89.7	61.1
Res3D [27]	Sports-1M	85.8	54.9
TSN [32]	ImageNet + Kinetics	91.1	-
I3D [3]	ImageNet + Kinetics	95.6	74.8
ARTNet [31]	Kinetics	93.5	67.6
T3D [6]	Kinetics	91.7	61.1
ECO [42]	Kinetics	94.8	72.4
PAS-Net	Kinetics	94.8	73.8

TABLE IV

COMPARISON WITH ECO ON UCF101 AND HMDB51.

Network	Frames	Accuracy(%) UCF101	HMDB51
ECO [42]	8	90.2	63.3
	16	91.6	68.2
	32	93.1	68.3
PAS-Net	8	91.4	68.8
	16	93.2	**71.6**
	32	**94.1**	70.1

TABLE V

ACCURACY COMPARISONS WITH RECENT WORKS UTILIZING POSE

Network	Optical flows	Accuracy(%) UCF101	HMDB51
Attention Pooling [11]		-	52.2
Chained [41]	✓	76.1	69.7
PoTion [5]		65.2	43.7
I3D + PoTion [5]	✓	98.2	80.9
PA3D [37]		-	55.3
I3D + PA3D [37]	✓	-	82.1
PAS-Net		94.8	73.8

datasets. Moreover, we conduct comprehensive experiments to do further comparisons with ECO in terms of utilizing different frames, *i.e.*, 8,16,32. As shown in Table. IV, benefited from leveraging pose information for action recognition, our PAS-Net consistently surpasses ECO on both UCF101 and HMDB51 across arbitrary frames. And our PAS-Net slightly

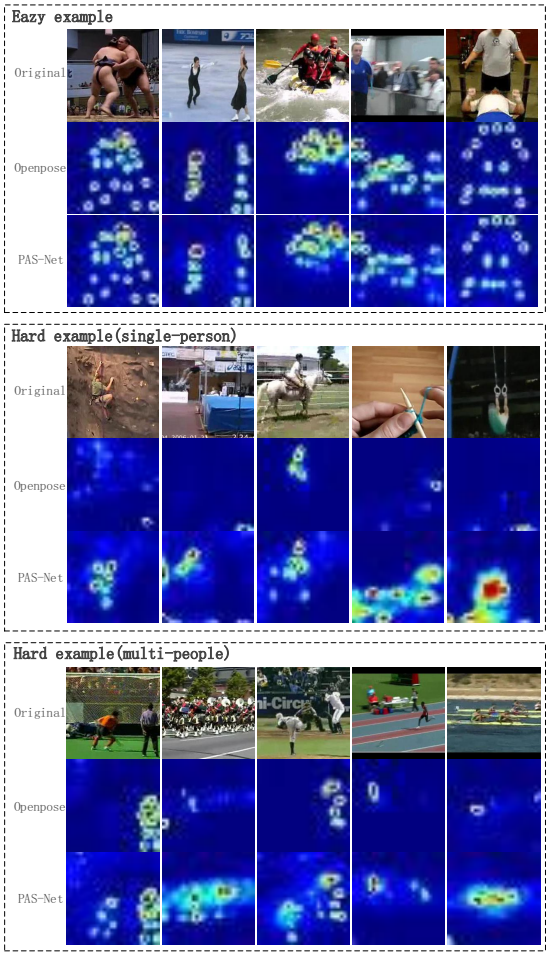


Fig. 3. Visualization comparison between Openpose and our PAS-Net.

underperforms I3D [3] since the latter employs a heavier network architecture, which is computationally expensive. Our PAS-Net with 32 frames performs 3x faster than I3D. Furthermore, we make comparisons with recent pose-related works [5], [11], [37], [41] in Table. V. Both PoTion [5] and PA3D [37], utilize heavy I3D as auxiliary architectures. However, neither exploiting auxiliary information like optical flow nor heavy network like I3D, our PAS-Net achieves competitive performance on both UCF-101 and HMDB-51, which demonstrates the effectiveness of our PAS-Net.

D. Visualization and Analysis

Although PAS-Net employs the pretrained Openpose model as the initializations to leverage the pose information for action recognition, the weights of the Openpose model are further optimized to improve the action recognition accuracy under an end-to-end training manner. Visualization results of pose heatmaps from the original Openpose model

and the modified versions in PAS-Net are shown in Fig. 3. It can be concluded that for easy examples, similar results are obtained for both the original Openpose and the version optimized by our PAS-Net. While for hard examples, our PAS-Net demonstrates the great advantages of well exploring attentions useful for action recognition, especially in small objects and the crowd. Interestingly, our PAS-Net pays attentions to not only persons but also objects interacting with people and we believe our PAS-Net seeks action-related pose heatmaps to improve the action recognition results.

V. CONCLUSIONS

We present a novel pose-based and appearance-based spatiotemporal networks(PAS-Net), which simultaneously consider the appearance information and pose knowledge. Moreover we explore various architectures for fusing the appearance and pose information, and our proposed training strategy can effectively reduce the influence of overfitting for the limited training data. Furthermore, our PAS-Net achieves competitive performance on UCF-101 and HMDB-51 when using only RGB modality. Future work includes investigating the effectiveness of our PAS-Net on other action analysis tasks such as action detection and video captioning.

REFERENCES

[1] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[4] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, 2015.

[5] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *CVPR*, 2018.

[6] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Amir Hossein Karami, Mohammad Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3d convnets: New architecture and transfer learning for video classification. *arXiv:1711.08200*, 2017.

[7] Wenbin Du, Yali Wang, and Yu Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *ICCV*, 2017.

[8] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[10] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019.

[11] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NIPS*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[14] Mohamed E Hussein, Marwan Torki, Mohammad A Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, 2013.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.

[16] Mihir Jain, Jan C Van Gemert, and Cees GM Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015.

[17] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013.

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.

[19] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[20] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *CVPR*, 2018.

[21] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. *arXiv:1903.06946*, 2019.

[22] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.

[23] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[24] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012.

[26] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[27] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv:1708.05038*, 2017.

[28] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[29] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. 2011.

[30] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[31] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, 2018.

[32] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[34] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[35] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. 2018.

[36] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. Multi-stream multi-class fusion of deep networks for video classification. In *ACM MM*, 2016.

[37] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *CVPR*, 2019.

[38] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[39] Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation? In *BMVC*, 2011.

[40] Jiagang Zhu, Zheng Zhu, and Wei Zou. End-to-end video-level representation learning for action recognition. In *ICPR*, 2018.

[41] Mohammadreza Zolfaghari, Gabriel L Oliveira, Nima Sedaghat, and Thomas Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *ICCV*, 2017.

[42] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, 2018.