Local fusion networks with chained residual pooling for video action recognition

He, Feixiang; Liu, Fayao; Yao, Rui; Lin, Guosheng

2018

He, F., Liu, F., Yao, R., & Lin, G. (2019). Local fusion networks with chained residual pooling for video action recognition. *Image and Vision Computing*, 81, 34-41.
doi:10.1016/j.imavis.2018.12.002

<https://hdl.handle.net/10356/143069>

<https://doi.org/10.1016/j.imavis.2018.12.002>

© 2019 Elsevier B.V. All rights reserved. This paper was published in *Image and Vision Computing* and is made available with permission of Elsevier B.V.

Downloaded on 26 Apr 2023 21:43:28 SGT

Local Fusion Networks with Chained Residual Pooling for Video Action Recognition

Feixiang He^a, Fayao Liu^{b,*}, Rui Yao^c, Guosheng Lin^a

^a*School of Computer Science and Engineering, Nanyang Technology University, Singapore.*

^b*School of Computer Science, The University of Adelaide, SA, Australia.*

^c*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China.*

Abstract

Action recognition is an important yet challenging problem. We here present a novel method, multistage local fusion networks with residual connections, to boost the performance of video action recognition. In realistic videos, an action instance may have a long time span and some frames may suffer from deteriorated object appearance due to motion blur or video defocus. Our method enhances the per-frame representation by capturing information from neighboring frames. We propose a local fusion block which considers neighboring frames to capture appearance and local motion information for generating per-frame representation. Our local fusion is performed in a multi-stage manner allowing feature fusion from varying neighborhood sizes in the temporal dimension. We employ residual connections in the fusion blocks to enable effective gradient propagation through the whole network allowing effective end-to-end training. We achieve competitive results on two challenging and public available datasets, namely HMDB51 and UCF101, which shows the effectiveness of the proposed method.

Keywords: action recognition, residual connection, local fusion, deep convolutional network.

1. Introduction

A rapid increase in the number of video acquisition devices (e.g., smart phones, surveillance cameras, etc.) has resulted in an enormous quantity of videos which surpasses the amount a single human can watch. Therefore, it has been a pressing need to develop automatic video understanding and analysis algorithms for various applications, such as action recognition.

**Corresponding author: Fayao Liu*

Email addresses: `fxhe@ntu.edu.sg` (Feixiang He), `fayaoliu@gmail.com` (Fayao Liu), `ruiyao@cumt.edu.cn` (Rui Yao), `gslin@ntu.edu.sg` (Guosheng Lin)

Figure 1: An illustration of the proposed approach. For an input sequence, our multistage aggregation is able to enhance the features of each frame by capturing the context in neighboring frames, and at the same time the local motion information is also captured in the frame features. The feature aggregation is implemented via a chained residual pooling module which is illustrated in Figure 3. The figures throughout this paper are best viewed in colors.

Action recognition has attracted significant research attention, and much progress has been made in methods using hand-crafted features [1, 2, 3, 4] and deep network based methods [5, 6, 7, 8, 9, 10]. Video action recognition is generally regarded as a classification problem, and thus it is essential to design a network to learn effective representations which can preserve the categories information of action classes. Recent deep ConvNets based approaches including ConvNet+LSTM approaches [11, 12, 13], 3D ConvNets [14, 5, 7, 15], and two-stream approaches with spatial and temporal networks [6, 8].

A number of existing approaches attempt to capture spatial-temporal pattern with the input of a stack of consecutive video frames. Representative approaches in this theme are based on 3D ConvNets. These approaches tend to be computational expensive due to the dense convolution along the temporal dimension, which may bring difficulties for training and prediction on long videos.

Two-stream based approaches have shown outstanding performance for action recognition. These methods typically consist of a spatial network and a temporal network. The spatial network is able to encode the information of scenes and objects present in the video, while temporal network encodes short-term motion information. The inputs of these networks are video frames and a stack of optical flow images, respectively. Our method is an extension of the two-stream method with additional stream for capturing simultaneously spatial

Figure 2: An illustration of our three-stream approach which consists of three parallel subnetworks, namely spatial network(top), temporal network(middle) and local fusion (LF) network (bottom), respectively. As in [8], an input video is divided into T segments, and a single frame or a stack of optical flow images is selected from each segments. The final video-level prediction is obtained by aggregating the label scores of three stream networks.

and temporal information in neighboring frames.

We propose an approach, residual local fusion network (LF network), to improve the accuracy of action recognition. The proposed method is able to enhance the features of a single video frame by leveraging the temporal context and local motion information from neighboring frames. A brief illustration of our approach is shown in Figure 1.

In realistic videos, an action clip may last for a few seconds and some frames may be in poor quality because of motion blur and video defocus. Our approach with multistage local fusion is able to enhance the features of blur frames and thus alleviate these issues. Moreover, representation learning from local neighboring frames in our method allowing implicit learning of local motion information, which are not well explored in existing two-stream based approaches.

As inspired by [16, 17], we exploit residual connections in the feature aggregation module of proposed approach. The design of residual connection strengthens its capacity of conveying gradients in the process of network parameter optimization.

Our method can be considered as an extension to two-stream networks. We combine our LF network stream with spatial and temporal streams as illustrated in Figure 2. The video-level class prediction is jointly determined by three network branches.

We perform experiments on two challenging publicly available video bench-

mark datasets, namely HMDB51 [18] and UCF101 [19], to verify the effectiveness of our method. Our method achieves the state-of-art performance on the dataset HMDB51 and UCF101.

To sum up, we make the following contributions.

- We propose a novel approach for video action recognition which is able to consider local temporal context to generate per-frame feature representation. We introduce a multistage local fusion block to generate frame features encoding local temporal information of varying length contexts. A set of such frame features in a temporal order construct the video representation which retains the temporal structure of an input video.
- We employ residual connections in our local fusion block which allows effective gradient propagation through the deep networks. Our residual fusion block facilitate effective end-to-end network training.
- We conduct experiments on two challenging and publicly available datasets and demonstrate that (i) the proposed method performs best among several feature aggregation baselines. (ii) our three-stream network achieves state-of-art performance.

The remaining sections are organized as follows. Related work is discussed in section 2,. Section 3 describes our proposed methods. Section 4 presents experiment results and analysis, followed by conclusion in section 5.

2. Related work

2.1. ConvNets for Action Recognition

Previous methods based on deep ConvNets are related to our method. Recently, several attempts have been proposed in [20, 6, 7, 8] to exploit image-level appearance information together with temporal information for action recognition. Karpathy *et al.*[5] train a ConvNet with deep structure on a large dataset (Sports-1M). Their model is trained on individual images and thus cannot capture the motion information across frames. To remedy this issue, Simonyan *et al.* propose a two-stream approach in [6] that consists of a spatial and a temporal network steam. The spatial stream captures the image appearance information of a single frame, while the temporal steam performs learning on optical flow stacks which represent the motions of video frames. As an extension to two-stream networks, Wang *et al.* in [8] propose Temporal Segment Networks (TSN) which take multiple segments into consideration by averaging feature representation of segments.

There are some other attempts to capture temporal information other than the two-stream framework. Tran *et al.* in [7] explore 3D ConvNets to directly capture spatio-temporal features from a stack of image frames. Sun *et al.* in [21] propose a factorized spatio-temporal ConvNet which decomposes the 3D convolution into 2D spatial and 1D temporal convolution. Feichtenhofer *et al.* in [20] exploit 3D convolution and 3D pooling to learn spatial and temporal features

on RGB images and a stack of optical flows. This type of 3D ConvNet based methods may suffer from massive computational cost. Recurrent networks, such as LSTM[11], are employed in [12, 22] to capture temporal dynamics and dependencies between consecutive frames.

2.2. Feature aggregation for action recognition

Feature aggregation is widely explored in action recognition [22, 23, 24, 13, 21, 25] and video description [12, 26]. Mean (or average) pooling is a simple approach to aggregate features from video frames, which was utilized in TSN [8]. Girdhar *et al.*[27] propose a method ActionVLAD which introduces a new video representation for action classification by aggregating local convolutional features across the entire spatio-temporal extent of the input video. ST-VLMPF proposed in [28] introduces a super vector encoding method which encodes local deep features over the entire video and demonstrates a remarkable boost in accuracy. The work in [23, 13, 12, 26, 25, 20] use recurrent neural network (RNNs) to aggregate features from consecutive frames.

Feichtenhofer *et al.* in [29] presents a general ConvNet architecture for video action recognition based on multiplicative interactions of space-time features. This model operates on a single frame and its corresponding optical flow stack, and combines the appearance and motion pathways of a two-stream architecture by motion gating. Wang *et al.* in [30] attempt to learn spatio-temporal features by fusing convolutional networks spatially and temporally and shows improvement on accuracy for action recognition.

In contrast to existing methods, we introduce a multistage fusion block in our network with residual connections. Our multistage fusion is able to generate frame features encoding local temporal information in varying lengths. Moreover, the residual connections employed in our local fusion block enable effective gradient propagation through the deep networks and thus facilitate effective end-to-end network training. Our method serves as an extension of the two-stream framework to encode local temporal context from neighboring frames.

3. Proposed Approach

We propose a multistage local fusion (LF) network to model local temporal context from neighboring frames. The architecture of LF network is shown in Figure 3. Our LF network can be effectively optimized via end-to-end training. We describe the details as follows.

A given video is denoted by X . Following TSN method in [8], we firstly divide it into T segments or parts of equal duration. A frame x_t is randomly sampled from the corresponding t -th segment. These sampled frames are used to represent the input video. Then we can denote a video as

$$X = [x_1, \dots, x_t, \dots, x_T], \text{ where } x \in R^{224 \times 224 \times K}. \quad (1)$$

Figure 3: The architecture of proposed method. The temporal preserving Local Fusion Block is constituted by multiple chained residual pooling units. The input of this fusion block is T feature vectors and the dimension of one feature vector is M .

In the spatial stream, one frame is represented by an RGB image ($K = 3$), while in the temporal stream, it is represented by a stack of optical flow images extracted from neighboring frames ($K = 10$). Therefore, the training set can be denoted as

$$\mathcal{X} = \{(X_i, y_i)\}_{i=1}^N \subset \mathbb{R}^{224 \times 224 \times K \times T} \times \{1, \dots, C\}, \quad (2)$$

where X_i is a training video and y_i is its class label. Here C denotes the number of classes.

After feature extraction, the features of each video can be formulated as

$$\hat{X} = [\hat{x}_1, \dots, \hat{x}_T], \text{ where } \hat{x} \in \mathbb{R}^{M \times T}. \quad (3)$$

Here $\hat{x}_t$ represents features extracted from RGB image or a stack of optical flow images, and M indicates the feature dimension.

The proposed method is a deep convolutional neural network which constitutes a feature extraction module, a novel local fusion module and a classification module. For each video, we sample a sequence of T frames. After the feature extraction module, a local fusion module $g(\cdot)$ operates on the extracted feature vectors of frame sequence. This module is able to perform multistage fusion of neighboring context for a reference frame and thus to enhance the per-frame features. In addition, it also serves to encode local temporal motion information across frames. The output of this module is the final feature representation of a video, $F_{final} = g(\hat{X})$, which not only includes the information carried in individual frame, but also preserves the sequential ordering information among frames. The output of the whole network is a C -way Softmax layer, which outputs a probability distribution, S , over the C classes. We denote by $\delta(\cdot)$ an indicator function. The network is trained to minimize the cross-entropy loss, which can be written as:

$$\mathcal{L} = - \sum_{i=1}^N \sum_{j=1}^C \delta(j = y_i) \log \left[\frac{\exp(S_{i,j})}{\sum_{k=1}^C \exp(S_{i,k})} \right]. \quad (4)$$

3.1. Local Fusion Module with Chained Residual Pooling

This is the key module of proposed method which enhances the per-frame features by multistage assembling consecutive frames along with the motion path. As illustrated in Figure 3, a temporal preserving local fusion module consists of several residual pooling units which are consecutively chained together. The Residual Pooling Unit, an important component of this module, operates on feature vectors and fuse the information along with temporal dimension, followed by a 1D convolution layer which adjusts the pooled features for adaptation before the fusion into the main information path. The sliding window size and stride of the temporal pooling are both set to 2, and the filter size and stride of the 1D convolution are set to 3 and 1 respectively. After the operation of temporal pooling and 1D temporal convolution, the dimension of feature vectors keeps unchanged. Then, the feature vectors are fused to the original feature vectors by summation, which follows the design of residual connection in ResNet [16]. By chaining Residual Pooling units sequentially, we perform multistage fusion from neighboring frames to enhance the feature of per-frame and capture the local temporal dynamics. Finally, we concatenate feature vectors from frames to form a temporal preserving representation of the input video.

Residual Pooling is able to enhance the appearance information carried in RGB frames and model the local motion information from neighboring frames. We employ cascade multiple residual pooling units to perform multi-stage fusion which allows to capture information from a large temporal window on neighboring frames. The residual connection employed in our fusion block enable effective gradient propagation and thus achieve effective network training. More discussion of residual connections can be found in [16].

4. Experiments

We evaluate our proposed method in this section. We first introduce datasets along with the experimental protocol and evaluation metrics. We then provide the implementation details of our proposed approach. We conduct several experiments on standard datasets and compare with baseline methods and a number of recent methods to demonstrate the effectiveness of our proposed method.

4.1. Datasets

we conduct experiments on two challenging publicly available human action classification datasets, named HMDB51[18] and UCF101[19].

HMDB51 dataset contains around 6800 trimmed videos in total. This dataset covers a large variety of actions, including facial actions, object manipulations, body movement and human interactions. This dataset is regarded as a challenging human action dataset due to the following aspects: (1) many videos in this dataset with poor quality and significant camera motions; (2) the number of samples is limited, which increases the risk of over-fitting especially when applying a deep network.

Method	Mean Pooling	Concat Feature	LF network
Split 1	54.31	57.39	59.08
Split 2	52.48	57.84	58.95
Split 3	54.77	58.43	59.93
Avg	53.85	57.89	59.28

Table 1: Comparison of different aggregation methods on HMDB51. Our method performs the best.

Method	Mean Pooling	Concat Feature	LF network
Split 1	85.38	86.31	86.44
Split 2	84.73	85.51	85.65
Split 3	85.09	85.50	85.61
Avg	85.07	85.77	85.9

Table 2: Comparison of different aggregation methods on UCF101. Our method performs the best.

UCF101 is a larger dataset and is often used for evaluation in previous work. It contains 13320 trimmed videos from 101 action classes that are divided into 5 categories, namely human-object interaction, body-movement only, human-human interaction, playing musical instruments and sports. In this dataset, variations in pose, camera motion, viewpoint and large spatio-temporal extent of an action bring the challenges for action recognition.

For these two datasets, we use the 3 training/testing splits provided in the original evaluation scheme and report the average accuracy over these 3 splits as their classification accuracy.

4.2. Implementation Details

We describe the implementation details of the proposed method, including training and testing phases. We use the Pytorch toolbox for ConvNets implementation.

Local fusion (LF) network:

As mentioned earlier in the previous section, the goal of our LF network is to capture long-range temporal structure over the video and strengthen feature representation by taking neighboring frames into consideration. We divide the video into 10 segments, and sample one snippet (frame) from each segment. The input of LF network is a stack of RGB frames whose size is $224 \times 224 \times 3 \times 10$.

Following the implementation in TSN [8], we employ the Inception Network with Batch-Normalization (BN-Inception) [31] to extract features, because of its good balance between accuracy and efficiency. To initialize the parameters of the network, we utilize a pre-trained model trained on the ImageNet dataset [32]. Since the category number of our dataset is different from that of Image-Net, we replace the last classification layer in the pre-trained model with a C -way Softmax layer, where C is the number of action categories in the dataset. Our local fusion module consists of two Residual Pooling units that are chained to

each other. The sliding window size of the temporal pooling and the filter size of the 1D convolution layer employed in a residual pooling unit are set to 2 and 3 respectively.

Training setting:

In the training phase, the network weights are learnt using the mini-batch stochastic gradient descent with momentum (set to 0.9) and weight decay (set to 5×10^{-4}). At each iteration, a mini-batch of 16 video samples is constructed by sampling 16 training videos (uniformly across the classes), from each of which ten frames are selected. Since the size of action recognition datasets, *e.g.* HMDB51 and UCF101, is small, training deep ConvNets on these datasets may lead to over-fitting. To alleviate this problem, we exploit several strategies, including regularization techniques and data augmentation techniques. As for regularization techniques, we choose to freeze the mean and variance parameters of all Batch Normalization layer except the first one, and add a dropout layer with drop ratio is 0.8. As for data augmentation, we utilize corner cropping (four corners and a center), scale-jittering [33] and horizontal flipping.

Testing setting:

The number of frames varies from video to video, we prepare our test input by uniformly dividing a video into 10 segments and randomly sampling a snippet (frame) from each segment. In other words, the input of our method is a sequence of RGB images with a length of 10. For video-level class prediction, we use 3 random samples of 10 frames extracted from the video, and use 5 crops (4 for corners and 1 for the center) of 224×224 along with their flipped version. In total there are 30 frame sequences for each video and we take the average result of all sequences for the final prediction of a video.

4.3. Ablation study

In this section, we demonstrate the effectiveness of LF network by (i) exploring different aggregation functions $g(\cdot)$ to assemble the segments; and (ii) combining LF network with other complementary networks for action recognition.

Aggregation methods:

For a fair comparison with other feature aggregation methods, we implement two baselines method using the same feature extraction module (BN-Inception) as in our LF network. The first baseline method is using Average Pooling for aggregation, and the second baseline method is using simple frame feature concatenation (referred to as Feature Concat) without aggregation. Average pooling is referred to mean operation across feature vectors, and Feature Concat is concatenating feature vectors into a single long vector and maintain the sequential ordering information. For the implementation of these two baselines, we basically replace our fusion module with Average Pooling or feature concatenation.

The performance of our method and the two baseline methods on HMDB51 and UCF101 are present in Table 1 and 2, respectively. In these experiments, 10 segments are sampled from each trimmed video. The base network BN-Inception is pre-trained on ImageNet. We use mini-batch stochastic gradient

Methods	HMDB51				UCF101			
	Split1	Split2	Split3	Average	Split1	Split2	Split3	Average
spatial(RGB)	54.77	51.05	50.59	52.14	86.55	85.16	84.66	85.46
LF network (RGB)	59.08	58.95	59.93	59.28	86.44	85.65	85.61	85.90

Table 3: Detailed comparisons between spatial network and the proposed LF network approach on the HMDB51 and UCF101 datasets.

Methods	HMDB51				UCF101			
	Split1	Split2	Split3	Average	Split1	Split2	Split3	Average
temporal (Flow)	63.53	63.99	65.23	64.25	87.73	90.14	90.99	89.62
spatial+temporal	71.24	68.37	70.39	70.00	93.95	94.19	94.53	94.22
spatial+temporal+ LF network	72.42	71.37	72.81	72.20	94.03	94.40	94.64	94.36

Table 4: Comparisons between two-stream approach [8] and the three-stream approach with out LF network on two datasets: HMDB51 and UCF101. By adding our LF network, our three-stream approach successfully improves the performance of the two-stream approach.

decent algorithm with momentum (set to 0.9) and weight decay (set to 5×10^{-4}) for network training. Meanwhile, the batch size is set to 16, and initial learning rate set to 0.001. As shown in Table 1 and 2, in terms of the average accuracy of three splits, Average pooling has the weakest performance, followed by Feature Concat and then our proposed method (53.85, 57.89, 59.28 in HMDB51 and 85.07, 85.77, 85.9 in UCF101). Our method performs the best.

Average Pooling baseline method is not able to capture local motion information and fail to preserve temporal structure of the input video. For the baseline method Feature Concat, it simply concatenate the feature vectors of frames into a single longer vector preserving the sequential ordering information of input frames, but this method is not able to explore the local context information of neighboring frames for generate the features of one frame. In contrast, our LF network performs multi-stage local fusion of features from neighboring frames to generate per-frame features, and thus we are able to capture the local context and local motions to strengthen the frame features. Concatenation of frame features in the final step of our approach preserves the temporal structure of the whole video.

Three-stream network:

Our LF network can be included in the existing two-stream network to form a three-stream approach. We utilise the models of a two-stream based method TSN [8] released by wang *et al.* to obtain the spatial steam and temporal steams. These two models are trained on Caffe platform with a setting where the batch size and the momentum are set to 256 and 0.9, respectively. For this trained model, the initial learning rate for the spatial stream is set to 0.001, while for the temporal network stream is set to 0.005. We transform these trained models to their Pytorch version, and test on Pytorch platform.

We first compare our LF network with the spatial stream of the method TSN [8]. Both networks use RGB images as input. The result is shown in Table 3. It shows that our LF network has better performance than the spatial network,

Model	HMDB 51	UCF 101
Factorized ConvNet[21]	59.1	88.1
Two-Stream + LSTM (GoogLeNet) [22]	-	88.6
Two-Stream (VGG-M) [6]	59.4	88.0
Two-Stream ConvNet [34] (VGG-16)	58.5	91.4
Two-Stream Fusion [20] (VGG-16)	65.4	92.5
TDD + FV [35]	63.2	90.3
RNN + FV [36]	54.3	88.0
LTC [37]	64.8	91.7
KVMF [38]	63.3	93.1
TSN (BN-Inception,2-modality) [8]	68.5	94.0
TSN+ (BN-Inception,2-modality)	70.4	94.2
Spatiotemporal Pyramid Network [30]	68.9	94.6
IDT+FV[2]	57.2	85.9
TDD + IDT [35]	65.9	91.5
C3D + IDT [7]	-	90.4
Two-Stream Fusion + IDT [20]	69.2	93.5
Dynamic Image Networks + IDT [39]	65.2	89.1
ST-ResNet + IDT [40]	70.3	94.6
Adascan + IDT + C3D [24]	66.9	93.2
ActionVLAD(VGG-16) + IDT [27]	69.8	93.6
ST-VLMPF + HMG + IDT [28]	73.1	94.3
Spatio-Temporal Multiplier Networks + IDT [29]	72.2	94.9
Ours	72.2	94.4
Ours + IDT	73.1	95.1

Table 5: Comparison of our method with existing methods on HMDB51 and UCF101. In the first part, we compare ConvNet-based methods using VGG-16 or comparable models. In the second part, we compare methods that use IDT [2]. In the last part, we report the performance of our proposed method.

which indicates our network with chained residual pooling is able to generate effective video representation contributing to video action recognition.

To further demonstrate the effectiveness of our proposed method, we construct a three-stream network which consists of three parallel subnetworks, namely spatial, temporal and LF network, as shown in Figure 2. The input of the spatial stream is RGB sequences from 3 segments, and the input of LF network is RGB sequences from 10 segments. For the temporal stream, the input is stacks of dense optical flow images from 3 segments, each stack is constituted by stacking optical flows in both X and Y directions from 5 neighboring frames. We extract optical flows using the tool provided by Wang *et al.* [8]. They use TV-L1 algorithm and quantize the optical flow fields in the range of [0,255] by a linear transformation.

In terms of video label prediction, we combine the prediction scores of these three sub-networks in a late fusion manner by averaging before Softmax operation, and yield the final prediction score of a video. Table 4 presents the detailed comparison of our proposed three-stream networks and the conventional two-stream network with spatial and temporal streams. The fusion weighting for spatial and temporal streams is set to 1 and 1.5 respectively. For the two-stream approach, we obtain an accuracy of 70% on HMDB51 and 94.22% on UCF101. Our three-stream approach, which is the combination of spatial, temporal and LF network, leads to a better performance, 72.2% on HMDB51 and 94.36% on UCF101. Here we set equal weights to spatial and LF network and weights for spatial, temporal, and LF is set to 1, 6 and 1, respectively.

Our LF network is designed to enhance the per-frame features. We perform local fusion by residual pooling to capture the local temporal context and local motions in neighboring frames. With this local fusion, it is expected that our LF network performs better than the simple spatial stream in the conventional two-stream approach. Our local fusion is also useful for those low-quality frames which are caused from large camera motion or defocus. In this case, our local fusion module transfers the features from neighboring frames to the low-quality frames.

We perform local fusion in a multi-stage manner, which help to capture the context information from different neighborhood sizes. Multi-stage fusion is also capable to capture long-range motion information by fusing information from a large neighborhood size.

The proposed LF network is complementary to the two-stream method, and including LF network in two-stream method improves the overall performance for action recognition.

4.4. Comparison with state-of-the-art

We compare our method with a number of recent methods in this section. Our three-stream approach achieves very competitive performance on both HMDB51 and UCF101 dataset. Following many existing methods, we employ IDT features to further improve the performance of our method.

Table 5 summarizes the results of our method and other comparing methods. We report the accuracy averaged over three splits, and all the methods are

	Video					TSN+	ours
HMDB51						run	kick_ball
	hug	swing_baseball
	fall_floor	stand
	shake_hands	ride_horse
	catch	flic_flac
	BreastStroke	BoxingPunchingBag
UCF101						Lunges	HighJump
	ParallelBars	PizzaTossing
	HorseRace	FloorGymnastics
	Knitting	Skijet

Figure 4: Prediction examples of our method and TSN. The test videos are from HMDB51 and UCF101 datasets. The last two columns show the prediction labels of TSN+ (BN-Inception, 2 modality) and our proposed method. Our method achieve correct predictions for these example videos.

trained without external labeled data. In the first part of the result table, we compare ConvNet-based methods using VGG-16 or comparable models; in the second part, we compare methods that use IDT [2] features; and lastly in the third part, we report the results of our proposed method using and without using IDT. Our method outperform most comparing methods in both datasets.

The result for "TSN (BN-Inception, 2 modality)" is the reported result of the TSN method in [8], and the method is implemented using Caffe toolbox. The result for "TSN + (BN-Inception, 2 modality)" is from the released Pytorch based models of the TSN. Combined with complementary IDT features and C3D method, the performance of AdaScan [24] achieves 93.2, 66.9 on the UCF101 and HMDB51 datasets, respectively. The fusion of Spatio-Temporal Multiplier Networks and IDT [29] produces a good performance of 94.9 on UCF101 dataset. ST-VLMPF [29] achieve competitive performance by fusing HMG and IDT. Our method integrates temporal network, spatial network and LF network, and gain a competitive results, 72.2 on HMDB51 and 94.4 on UCF101. Further fusing IDT feature, we gain best results on these two datasets – 73.1 on HMDB51 and 95.1 on UCF101.

4.5. Visualizations of examples

To qualitatively demonstrate the effectiveness of our method, We illustrate some prediction examples of our method and compare with TSN [8]. Our method capture local temporal context and local motions to enhance the frame features. Especially when the video with poor quality because of motion blur or video defocus, our method is still able to generate effective representation for video frames through our multi-stage local fusion.

Prediction examples are shown in Figure 4. The first 5 videos are from HMDB51 dataset, and the remaining are from UCF101 dataset. The prediction labels of the method TSN+ (BN-Inception,2 modality) are marked in red, and the prediction results our proposed three-stream based method (combining LF network , spatial and temporal network) are marked in Green. Our method manages to output correct prediction on these input videos.

5. Conclusion

We have presented an action recognition method which strengthens the feature representation of video frames by capturing local temporal context. We introduce chained residual pooling to perform multistage fusion of neighboring frames. The employed residual connections in our fusion blocks enable effective gradient propagation and thus help network training. We combine our LF stream with spatial and temporal streams to form a three-stream network which is an extension of the conventional two-stream framework for action recognition. As evaluated on two challenging datasets for human action, namely HMDB51 and UCF101, we have demonstrated the effectiveness of our method and achieve very competitive performance.

Acknowledgment

We would like to thank NVIDIA for GPU donation.

References

- [1] I. Laptev, On space-time interest points, *International journal of computer vision* 64 (2-3) (2005) 107–123.
- [2] H. Wang, C. Schmid, Action recognition with improved trajectories, in: *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.
- [3] L. Wang, Y. Qiao, X. Tang, Motionlets: Mid-level 3d parts for human motion recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2674–2681.
- [4] L. Wang, Y. Qiao, X. Tang, Mofap: A multi-level representation for action recognition, *International Journal of Computer Vision* 119 (3) (2016) 254–271.
- [5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
- [6] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: *Advances in neural information processing systems*, 2014, pp. 568–576.
- [7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.
- [8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: *European Conference on Computer Vision*, Springer, 2016, pp. 20–36.
- [9] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: *European Conference on Computer Vision*, Springer, 2016, pp. 816–833.
- [10] J. Liu, G. Wang, P. Hu, L.-Y. Duan, A. C. Kot, Global context-aware attention lstm networks for 3d action recognition, in: *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 1647–1656.
- [11] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, A. Courville, Recurrent batch normalization, *arXiv preprint arXiv:1603.09025*.
- [12] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

- [13] Z. Li, K. Gavriluk, E. Gavves, M. Jain, C. G. Snoek, Videolstm convolves, attends and flows for action recognition, *Computer Vision and Image Understanding*.
- [14] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, *IEEE transactions on pattern analysis and machine intelligence* 35 (1) (2013) 221–231.
- [15] G. W. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional learning of spatio-temporal features, in: *European conference on computer vision*, Springer, 2010, pp. 140–153.
- [16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [17] G. Lin, A. Milan, C. Shen, I. Reid, Refinenet: Multi-path refinement networks for high-resolution semantic segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [18] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: *Computer Vision (ICCV)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 2556–2563.
- [19] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, *arXiv preprint arXiv:1212.0402*.
- [20] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.
- [21] L. Sun, K. Jia, D.-Y. Yeung, B. E. Shi, Human action recognition using factorized spatio-temporal convolutional networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4597–4605.
- [22] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.
- [23] S. Sharma, R. Kiros, R. Salakhutdinov, Action recognition using visual attention, *arXiv preprint arXiv:1511.04119*.
- [24] A. Kar, N. Rai, K. Sikka, G. Sharma, Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos, *arXiv preprint arXiv:1611.08240*.
- [25] N. Ballas, L. Yao, C. Pal, A. Courville, Delving deeper into convolutional networks for learning video representations, *arXiv preprint arXiv:1511.06432*.
- [26] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4507–4515.

- [27] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, B. Russell, Actionvlad: Learning spatio-temporal aggregation for action classification, arXiv preprint arXiv:1704.02895.
- [28] I. Cosmin Duta, B. Ionescu, K. Aizawa, N. Sebe, Spatio-temporal vector of locally max pooled features for action recognition in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3097–3106.
- [29] C. Feichtenhofer, A. Pinz, R. P. Wildes, Spatiotemporal multiplier networks for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4768–4777.
- [30] Y. Wang, M. Long, J. Wang, P. S. Yu, Spatiotemporal pyramid network for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1529–1538.
- [31] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
- [32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.
- [33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
- [34] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, Towards good practices for very deep two-stream convnets, arXiv preprint arXiv:1507.02159.
- [35] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4305–4314.
- [36] G. Lev, G. Sadeh, B. Klein, L. Wolf, Rnn fisher vectors for action recognition and image annotation, in: European Conference on Computer Vision, Springer, 2016, pp. 833–850.
- [37] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [38] W. Zhu, J. Hu, G. Sun, X. Cao, Y. Qiao, A key volume mining deep framework for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1991–1999.
- [39] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, S. Gould, Dynamic image networks for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3034–3042.
- [40] C. Feichtenhofer, A. Pinz, R. Wildes, Spatiotemporal residual networks for video action recognition, in: Advances in Neural Information Processing Systems, 2016, pp. 3468–3476.