

Temporal-Spatial Mapping for Action Recognition

Xiaolin Song, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jingyu Yang, and Xiaoyan Sun

Abstract—Deep learning models have enjoyed great success for image related computer vision tasks like image classification and object detection. For video related tasks like human action recognition, however, the advancements are not as significant yet. The main challenge is the lack of effective and efficient models in modeling the rich temporal spatial information in a video. We introduce a simple yet effective operation, termed Temporal-Spatial Mapping (TSM), for capturing the temporal evolution of the frames by jointly analyzing all the frames of a video. We propose a video level 2D feature representation by transforming the convolutional features of all frames to a 2D feature map, referred to as *VideoMap*. With each row being the vectorized feature representation of a frame, the temporal-spatial features are compactly represented, while the temporal dynamic evolution is also well embedded. Based on the *VideoMap* representation, we further propose a temporal attention model within a shallow convolutional neural network to efficiently exploit the temporal-spatial dynamics. The experiment results show that the proposed scheme achieves the state-of-the-art performance, with 4.2% accuracy gain over Temporal Segment Network (TSN), a competing baseline method, on the challenging human action benchmark dataset HMDB51.

Index Terms—Temporal-Spatial Mapping (TSM), action recognition, deep learning

I. INTRODUCTION

ACTION recognition is an important yet challenging problem in computer vision, with many practical applications such as visual surveillance, human computer interaction, and video analyses [1]. Recently, deep learning models like Convolutional Neural Networks (CNN) [2]–[5] and Recurrent Neural Networks (RNN) [6]–[11] have been extensively employed to recognize actions in videos. Despite great efforts and rapid developments, the advancements are not as significant as those achieved in image related computer vision tasks such as image classification [12]–[14] and object detection [15]–[17]. The main reason is that actions in a video involve not only the spatial information of each frame, but also its temporal evolution. Exploring this rich temporal-spatial information requires the deep learning model to be equipped with more parameters, trained with more video samples, and most importantly formulated with more effective architecture.

Previous attempts to address action recognition include the two-stream ConvNets [2], [3], [5], [18], 2D ConvNets followed by Long Short-Term Memory (LSTM) networks [6], [7], [9]–[11], 3D ConvNets [19], [20], and many others [8], [21], [22]. These models are continually pushing forward the state-of-the-art performances of action recognition. Most of these models, however, suffer from limitations such as lack of joint temporal-spatial learning [2], [3], [5], [18], difficulties in model training [6], [7], [9], [19], [20]. Moreover, limited by the designs, most of those approaches cannot leverage more dense frames for obtaining further gains even though more frames can provide more temporal-spatial information [3], [23], [24]. That

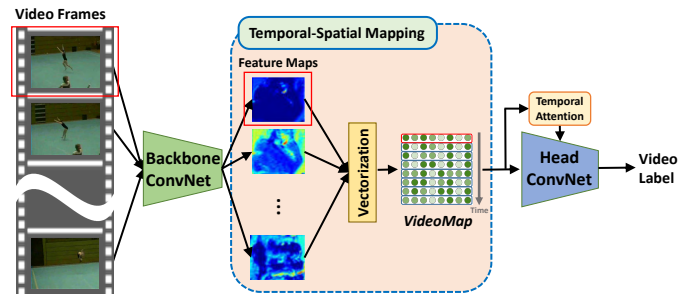


Fig. 1. Overview of our network structure. The Temporal-Spatial Mapping operation enables the effective joint temporal-spatial modeling by representing the temporal spatial features of an entire video by a 2D *VideoMap*. A temporal attention model in a head ConvNet further transforms the *VideoMap* to a compact video-level feature embedding for classification.

is because the statistics of the features/scores of frames are utilized to get the final prediction [3], [23]. Frames above a certain number (*e.g.*, 25 frames) help very little once they are enough to estimate the statistics. To overcome these limitations, we need a network architecture which jointly and effectively learns the temporal-spatial feature representations and is capable of exploring the information of dense frames.

To this end, we present a simple yet effective operation, *i.e.*, Temporal-Spatial Mapping (TSM), for joint temporal-spatial feature modeling. We represent the temporal-spatial features of an entire video compactly by a *VideoMap*, which is a row-wise layout of the per-frame vectorized ConvNet features as illustrated in the middle of Figure 1. This enables the “seeing” of dense frames at a glance and thus performing effective joint temporal-spatial analyses. The proposed TSM operation is general and can be used after any convolutional features for video-level temporal-spatial feature learning.

To deploy this TSM operation for action recognition, we first train a backbone 2D ConvNet model to extract convolutional features for each frame of a video sequence, then the TSM operation is performed on the features to generate the temporal-spatial *VideoMap*, which naturally encodes the temporal-spatial information in 2D feature map. Based on the compact *VideoMap* representation, we further propose a temporal attention model within a head ConvNet to extract effective video-level feature embeddings to predict the final action categories. Experiment results on two large benchmarks, HMDB51 and UCF101, demonstrate the effectiveness of the proposed network architecture and its state-of-the-art performances.

To summarize, the main contributions of this work are three-fold:

- We propose a simple yet effective operation, Temporal-Spatial Mapping, for jointly embedding the temporal-spatial information of a video from per-frame features into a compact feature map, *i.e.*, *VideoMap*. The pro-

posed operation is general and can be applied to any CNN features to explore temporal dynamics. *VideoMap* representation provides a way to leverage dense frames for enhancing the performance.

- We propose a temporal attention model within a head ConvNet to further transform the temporal-spatial *VideoMap* to a more compact and effective video-level feature representation for classification, which can better exploit the temporal-spatial dynamics.
- We present a deep architecture for action recognition which achieves significant performance improvement on the HMDB51 dataset. The source code and trained model will be released to facilitate the research in action recognition.

II. RELATED WORK

Motivated by the outstanding performance of deep neural networks on image classification and detection, more and more works have extended CNN-based or CNN-LSTM-based architectures for video analysis.

The two-stream ConvNets approaches [2], [3] train two separate 2D ConvNets for both appearance in still images and stacks of optical flow, from several sparsely sampled frames. The temporal stream takes the short-term temporal information into account by means of optical flow and achieves superior performance than the spatial stream (still images). The Temporal Segment Networks (TSN) [3] combine a sparse temporal sampling strategy and video-level supervision in training to explore temporal structure. Three frames which are randomly selected from three equally divided segments are jointly trained with frames combined by average pooling. However, these approaches involve temporal information by simply averaging/multiplying the scores across frames for the video level prediction. Such approaches still cannot accurately model the temporal dynamics [2], [3], [23]. Some aggregation techniques like VLAD [25], Fisher Vector [26] and dictionary learning [27] have been used in action recognition for aggregating features of frames [28]–[30]. VLAD-based methods like ActionVLAD [28] provide solutions to perform spatio-temporal aggregation of a set of action primitives. However, they do not model the time order of frames.

To extend convolution operations from 2D image to 3D video, the 3D ConvNets [20] directly operates on the video for spatio-temporal feature learning by replacing 2D filters with 3D ones. So far, such approach however has shown limited benefit, probably due to the lack of training data, high complexity of training 3D convolution kernels, and not exploiting the optical flow stream explicitly. To reduce the number of parameters of 3D ConvNets, Sun *et al.* [31] propose to factorize the original 3D convolution kernel learning as a sequential process of learning 2D spatial kernels followed by learning 1D temporal kernels. However, both approaches model the temporal dynamics by averaging the activations of the sub-clips of a video. There is still a lack of a global modeling of the action among the video sub-clips. To capture the relationships among the video sub-clips, the temporal linear encoding (TLE) [23] encodes the aggregated information of K (*i.e.*, $K = 3$) clips/frames into a video feature

representation by performing element-wise multiplication of the features of the clips/frames. However, for a long video, the sampling of K clips/frames and the aggregation of them by multiplication results in information loss. Since the statistics of clips/frames are explored rather than their details, like TSN [3], the performance increases little when more frames are used.

In [6]–[9], LSTM is utilized to explore the temporal evolution of the per-frame CNN features across a video. Shi *et al.* [32] propose a sequential Deep Trajectory Descriptor (sDTD) to model long-term motion information in video and employ a three-stream CNN-LSTM architecture for action recognition. Wang *et al.* [33] propose two-stream 3D ConvNets Fusion to recognize actions of arbitrary size and length by using spatial temporal pyramid pooling (STPP) with a LSTM or CNN model to extract multi-size descriptions and learn global representation for each input video. Li *et al.* [34] propose a unified Spatio-Temporal Attention Networks (STAN) using attention neural cell (*AttCell*) based on CNN-LSTM architecture to estimate attention on both spatial and temporal locations in a video. Compared with image-based approaches [2], [3], LSTM-based approaches go one step further which can model the temporal dynamics of a video. However, applying the LSTM models to video based action recognition has so far only achieved similar performance as temporal pooling [6], likely attributed to the rigid structure of LSTM and the difficulties in training.

In this paper, we propose a general approach of temporal-spatial mapping to facilitate the joint analysis of the dense frames/clips of a video, with the time order information embedded in the mapped *VideoMap*. Our approach provides an efficient way to explore the details of dense frames, enabling performance improvement.

III. TEMPORAL-SPATIAL MAPPING OPERATION

In a video, besides the spatial information in each image, the temporal evolution provides vital information for identifying an action. It is not trivial to find a video representation that encodes the dense frames together to facilitate the joint analysis of the entire video. The traditional powerful approach iDTs [24] densely samples feature points in video frames and uses optical flow to track them to yield a good video representation. For action recognition from video, most deep learning based approaches which have good performance [2], [3], [18] are still image based where they infer the final results by averaging/multiplying the scores/features of frames. The proposed Temporal-Spatial Mapping provides a way to jointly consider the dense sequential frames for inferencing the video label.

Figure III shows the process. For each frame of a video sequence, ConvNet generates feature maps at each convolutional layer, where features of higher abstraction are captured by higher layers [35]. Consider the output features of 2D ConvNet for T frames from a video. The features of the k^{th} frame can be a feature vector of L -dimension, *i.e.*, $f_k \in \mathbb{R}^L$, *e.g.*, the output of the global pooling layer of the TSN [3]. They can be feature maps with high dimensions, *i.e.*, $S_k \in \mathbb{R}^{h \times w \times c}$, where

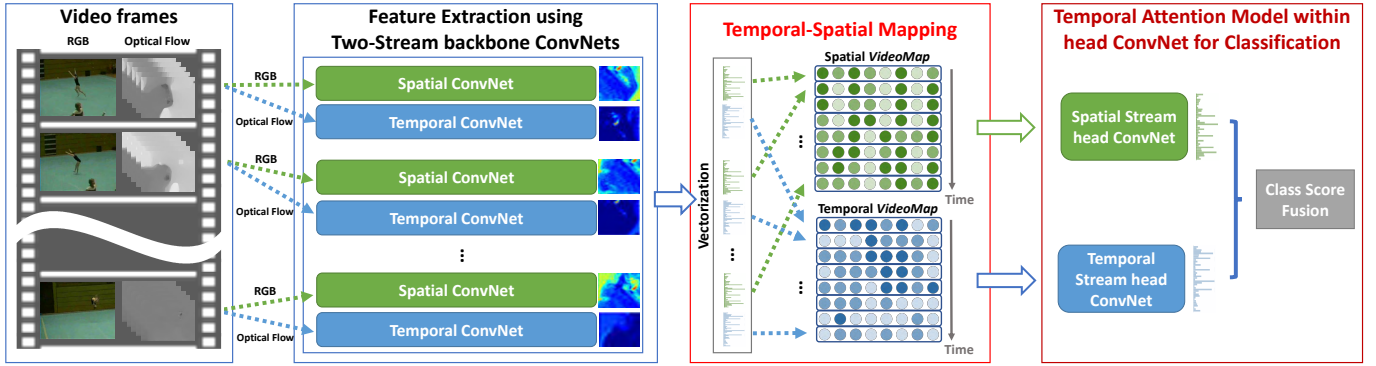


Fig. 2. The overall framework with our Temporal-Spatial Mapping operation followed by a head ConvNet for action recognition. Two-stream ConvNets extract features on each frame for the spatial stream (RGB) and temporal stream (Optical flow), respectively. The vectorized feature vectors of the sequential frames form a *VideoMap* for temporal-spatial representation. A head ConvNet with temporal attention makes action classification based on the *VideoMap*. Finally the class scores of the *VideoMaps* from two streams are fused to produce the video-level prediction.

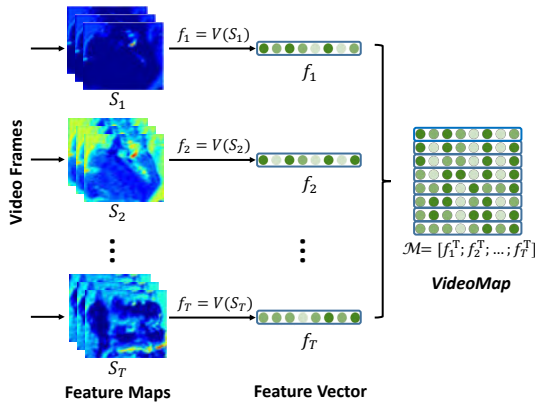


Fig. 3. Illustration of the proposed Temporal-Spatial Mapping operation, which transforms a sequence of feature maps into a compact *VideoMap*.

h , w and c denote the height, width, and number of channels of the feature maps, *e.g.*, the inception layer output of TSN [3]. The high dimensionality of feature maps makes it challenging to jointly analyze the dense frames of a video. Thus, in that case, a spatial vectorization function $V : S_k \rightarrow f_k$, is used to encode the feature maps to a low dimension vector of fixed length, *i.e.*, $f_k = V(S_k)$, where $f_k \in \mathbb{R}^L$. Then, we layout the feature vector of each frame as a row with the row identity corresponding to the time order of the frames to create a two-dimensional temporal-spatial map, *i.e.*, *VideoMap* as

$$\mathcal{M} = [f_1^T; f_2^T; \dots; f_T^T] \in \mathbb{R}^{T \times L}. \quad (1)$$

The width of the map is equal to the total number of frames while the height is equal to the dimension of the feature vector. The map has embedded both the temporal and spatial information. This makes it possible to have a global observation of a video sequence and facilitates the exploration of the temporal dynamics.

This TSM operation has the following characteristics and advantages.

- This TSM operation is a general operation which can be applied to the feature maps/features from ConvNets for encoding the temporal-spatial dynamics from a sequence of frames.

- This TSM operation for obtaining a *VideoMap* can maintain the time order information of the dense frames, which helps distinguishing action categories related with occurrence order, *e.g.*, “stand up” versus “sit down”.
- This TSM operation is simple yet effective. It does not involve complicated operations.

Discussions. Here, we discuss the relation and differences of our method with several classical methods. We aim to explore the temporal dynamics from the dense frames of a video and jointly make a decision from them.

Two-stream based ConvNets: Our framework is compatible with the two-stream ConvNet based approaches [2], [3]. TSN [3] uses three temporal segmented frames to explore the long-range temporal structure in training. During testing, the scores from N ($N = 25$) frames are averaged to finally predict the action. However, the temporal dynamics are only weakly explored by the simple averaging of a few frames without considering the time order. In contrast, our temporal-spatial mapping to a *VideoMap* enables the joint exploration of many frames with time order retained.

3D Convolution: 3D convolution provides an elegant framework for exploring the spatial and temporal dynamics [19], [20]. Without the practical constraints, such as on memory, labeled data, computational resource, it is expected to achieve excellent performance. However, such approaches so far have not demonstrated satisfactory performance in practice due to the difficult to train. In practice, 3D Convolution only covers a short range of the sequence for each input video sub-clip (*e.g.*, 5-7 frames in [19], 16 frames in C3D [20]). The scores of sub-clips are averaged to get the final prediction. The temporal dynamics among clips are not well explored by the simple averaging and the time order information among the sub-clips is lost.

CNN+LSTM: To tackle the not well solved temporal dynamic modeling problem, some works [6]–[9] use the Recurrent Neural Network with Long-short Term Memory (LSTM) to model the temporal evolution. The RNN structure facilitates the exploration of temporal dynamics from the dense frames with time order considered. However, it has only achieved similar performance as temporal pooling [6]. This might be attributed to the difficulty in training with the gradient

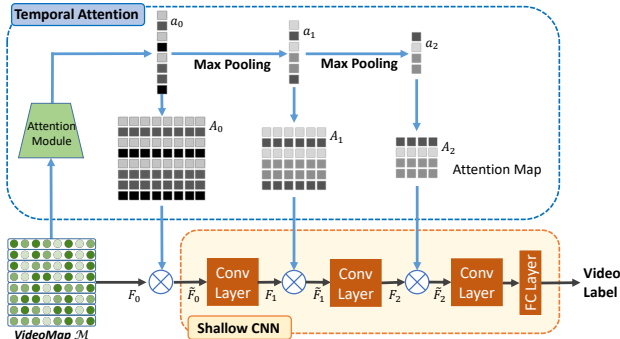


Fig. 4. Temporal attention model within a head ConvNet for action recognition from *VideoMap*.

vanishing for the long videos.

In contrast, we leverage the temporal-spatial mapping to obtain a *VideoMap* which embeds the information of temporal dynamics and time order. It facilitates the joint exploration of the dense frames of a video for a global decision.

IV. TEMPORAL ATTENTION WITHIN HEAD CONVNET

Motivated by the success of using ConvNet for feature extraction in image classification, which jointly explores cross-pixel correlations, we leverage a convolution neural network to jointly explore the cross-frame dynamics.

With the *VideoMap* as input, we design a temporal attention model within a head ConvNet for video level feature extraction and action recognition. Figure IV shows this network structure, which consists of a shallow ConvNet and a temporal attention module. Note that we refer to this shallow ConvNet as head ConvNet since it is the last sub-network specific to the task [36]. The responses from the temporal attention module are incorporated into the head ConvNet to adjust the importance level of temporal features. Cross-entropy as used in [3] is taken as the video level loss function.

To recognize the action class in a video, the importance level of different frames differs. Some frames are more likely to be irrelevant or less relevant to the action category and may hurt the final performance by introducing noise. Some other frames are more relevant to the action category. Take the action of handshake as an example, the frames with two people approaching are less relevant to the action, which could be shared by other action types, while the frames with two people's hands holding together give more discriminative information. Therefore, we introduce a temporal attention model for learning and determining the importance levels.

As illustrated in Figure IV, the feature maps \tilde{F}_i of the i^{th} layer of the head ConvNet after enforcing the temporal attention are described as:

$$\tilde{F}_i = A_i(\mathcal{M}) \circ F_i(\mathcal{M}), \quad (2)$$

where $F_i(\mathcal{M})$ denotes the output feature maps of the i^{th} layer of the head ConvNet, $A_i(\mathcal{M})$ is the attention map from the attention model, \circ denotes entrywise product. Here $A_i(\mathcal{M}) = a_i(\mathcal{M}) \otimes \mathbf{1}$, where \otimes is the Kronecker product and $\mathbf{1}$ denotes the all-ones vector, $a_i(\mathcal{M})$ is the learned attention vector with the dimension being related with the total number of frames.

In other words, $A_i(\mathcal{M})$ is the column-wise repeat of $a_i(\mathcal{M})$. Note that for the 0^{th} layer, $F_0(\mathcal{M}) = \mathcal{M}$, $a_0(\mathcal{M}) \in \mathbb{R}^T$. $a_i(\mathcal{M})$ is the max pooling result of $a_{i-1}(\mathcal{M})$ with a stride of 2. The detailed network designs will be described in Section VI.

V. OVERALL FRAMEWORK

We take the two-stream based ConvNets as our backbone ConvNets and embed the proposed TSM operation followed by a head ConvNet with temporal attention model, for the video-level classification. Figure 2 shows the overall flowchart of our final framework.

Temporal Segment Networks (TSN) [3] with BN-Inception structure [37] provides superior performance on both the spatial and temporal streams. We take the TSN as our backbone ConvNets for frame feature extraction. The network contains two streams: spatial stream with RGB image as input, and temporal stream with optical flow as input. The results from the two streams are fused to predict the video label.

Without loss of generality, we take the spatial stream as example to describe our overall network structure. The temporal stream acts similarly. For successive of frames in a video, the backbone spatial ConvNet outputs feature maps for each frame. With the feature maps of each frame vectorized to a feature vector, the feature vectors of the successive frames are arranged row-by-row to form a *VideoMap*. The *VideoMap* goes through the head ConvNet with temporal attention and the class scores are generated. The Temporal-Spatial Mapping operation permits the end to end training of the entire network. Due to memory constraints, in practice, we train the networks in two stages. In the first stage, we train the backbone ConvNets. Then we train the the head ConvNet for *VideoMap* classification.

VI. EXPERIMENTS

We validate the effectiveness of the proposed framework on two benchmark datasets. We first describe the datasets and implementation details. Then we study the effects of different factors in our network. Finally, we compare our approach with many state-of-the-art approaches.

A. Datasets

We conduct experiments on two popular human action recognition datasets, namely HMDB51 [38] and UCF101 [39]. The HMDB51 dataset is a very challenging dataset with higher intra-class variations and smaller inter-class variations. The videos are collected from movies and a variety of YouTube consumer videos. This dataset consists of 6,766 video clips from 51 action categories, with each category containing at least 100 clips. In each split, each action class has around 70 clips for training and 30 clips for testing. The UCF101 dataset consists of 13320 video clips in 101 categories. This dataset provides large diversity in terms of actions, variations in background, illumination, camera motion and viewpoints, as well as object appearance, scale and pose.

B. Implementation Details

Two-stream Backbone ConvNets. We pre-train our backbone network of the TSN the same way as reported in [3]. The size of input images or optical flow stacks is fixed at 256×340 . In order to avoid over-fitting, we perform data augmentation, by cropping images with the width and height chosen from four different sizes, 256, 224, 192 and 168, from five spatial locations of the full image, *i.e.*, one center and four corners. These cropped regions will be resized to 224×224 for feature extraction. Note that, to maintain spatial consistency across a video, the cropped size and location are the same across a video sample.

Temporal-Spatial Mapping and head ConvNet. For the k^{th} frame, the top inception module of the TSN outputs feature maps S_k of size $h \times w \times c$, where $h = 7, w = 7$, and $c = 1024$. An average pooling on each channel vectorizes them to a 1024-dimension vector f_k . For T sequential frames, a *VideoMap* $\mathcal{M} = [f_1^T; f_2^T; \dots; f_T^T]$ is obtained.

Video level feature learning and classification based on the *VideoMap* is performed using our head ConvNet with temporal attention. We build the head ConvNet by stacking three convolution blocks. In each block, it consists of a convolutional layer with 5×5 kernels, a ReLU layer, followed by a pooling layer of 3×3 sized kernel of stride 2. We construct the temporal attention module by two such convolution blocks followed by a fully connected layer which outputs a T dimensional vector, representing the frame-wise attention responses for a video. Since the GPU memory is limited, we set the mini-batch to have 128 *VideoMaps*. We set the initial base learning rate to 0.01 and decrease it by a factor of 10 every 10,000 iterations. We stop the training process after 100 epochs.

Vectorization of Feature Maps. For the high level convolution feature maps, the average pooling (*e.g.*, BN-Inception [3]) or full connection (*e.g.*, AlexNet [2]) of the feature maps has been done in the ConvNets structure, to convert them to a low dimensional feature vector. Therefore, we directly utilize such feature vectors from the sequential frames to form a *VideoMap*.

Given the feature maps of high dimension from a ConvNet layer, if the feature maps are not already transformed to a feature vector, a module for vectorization of the feature maps is needed to convert the feature maps to a low dimensional feature vector representation. There are many ways to perform to perform the vectorization, *e.g.*, leveraging a ConvNet to encode the feature maps to a feature vector.

C. Ablation Study

In this subsection, we will analyze the effectiveness of the proposed Temporal-Spatial Mapping component, discuss the design of the head ConvNet, analyze the effectiveness of the temporal attention module, and the influence of the height of the *VideoMap* (*i.e.*, density of temporal sampling), respectively.

Effectiveness of Temporal-Spatial Mapping. Aggregation of single-frame features of dense frames to a *VideoMap* provides the opportunity to jointly explore the temporal-spatial dynamics at the video level. We show the performance of

TABLE I
COMPARISON WITH TWO-STREAM BASED NETWORKS IN ACCURACY (%) ON THE HMDB51 DATASET (SPLIT 1). HERE “TWO-STREAM” [2] USES VGG-16 AS THE NETWORK STRUCTURE, AND “TSN [3]” USES BN-INCEPTION. “TSN+TSM(OURS)” IS OUR SCHEME WITH THE TEMPORAL-SPATIAL MAPPING (TSM) FOLLOWED BY A HEAD CONVNET.

Method	RGB	Optical Flow	Fusion
Two-stream [5]	42.2	55.0	58.5
Two-stream+TSM (Ours)	43.1	56.2	60.3
TSN [3]	54.6	62.6	70.8
TSN+TSM (Ours)	55.0	63.1	72.4

our scheme in comparison with the baseline scheme on the HMDB51 dataset (Split 1) in Table I.

In Table I, “TSN [3]” denotes the results of the TSN approach [3] with the BN-Inception structure, serving as our baseline scheme. We take its ConvNets as our backbone network to extract frame level features. “TSN+TSM (Ours)” denotes our scheme with the Temporal-Spatial Mapping operation and a head ConvNet for the joint temporal dynamics exploration from densely sampled frames. We can see that our scheme achieves 1.6% after the two-stream fusion. Note that in Table I and Table II, TSN results are obtained from the original TSN model but ran based on a new Caffe version on Windows.¹

We also evaluate the performance when using another backbone network, which takes two-stream ConvNets with VGG-16 network structure [5] as the frame-level feature extractor. Similarly, TSM brings 1.8% improvement in accuracy.

Comparisons on Network Designs. We have designed a head ConvNet for encoding the *VideoMap* for action recognition. The purpose of this network is to explore temporal dynamics among frames to learn efficient video feature representation. For the head ConvNet with *VideoMap* as input, we have tried three network structures: AlexNet, our designed 3-layer ConvNet, and 1-layer ConvNet. In addition, one alternative way for temporal modeling of the feature vectors is to use the Recurrent Neuron Network with LSTM. However, RNN suffers from the gradient vanishing problem even though LSTM suffers less than RNN. We show the results of these designs in Table II with experiments conducted on the HMDB51 dataset (Split 1). We can see that our 3-layer ConvNet (*TSN+TSM (3-layer ConvNet)*) achieves much superior performance than the LSTM based network. The performance of AlexNet is inferior to the 3-layer ConvNet since a deeper network with more parameters is prone to over-fitting. The 1-layer ConvNet does not converge in modeling the temporal-spatial dynamics.

Effectiveness of Temporal Attention. Different frames generally have different importance levels for recognizing the action. The less relevant frames or irrelevant frames could be noise which hurts the final performance. We have designed a temporal attention module with hierarchical attentions as shown in Figure IV (which is Figure 3 in the paper) and found that the hierarchical structure provides superior performance.

¹The results reported in website (<http://yjxiong.me/others/tsn/#exp>, spatial stream, temporal stream and the final fusion are 54.4%, 62.4% and 69.5%, respectively) of the original TSN are a little different from our ran results due to the difference of Caffe version on Windows.

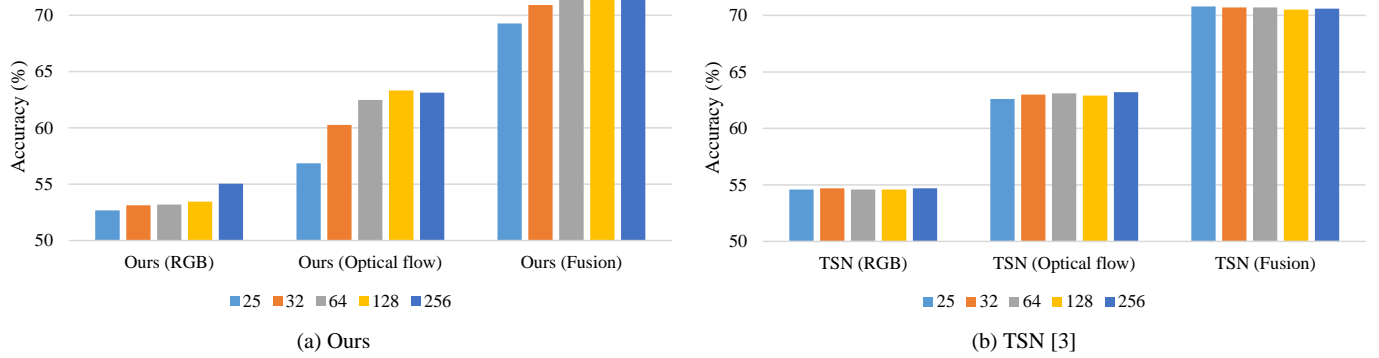


Fig. 5. Comparison of performance (accuracy %) at different frame sampling densities (number of frames: 25, 32, 64, 128, 256, respectively) for our method and the TSN on the HMDB51 dataset (Split 1). (a) Ours; (b) TSN [3]. In our scheme, the performance increases as the sampling density increases. In TSN, however, the performance does not increase as the sampling density increases.

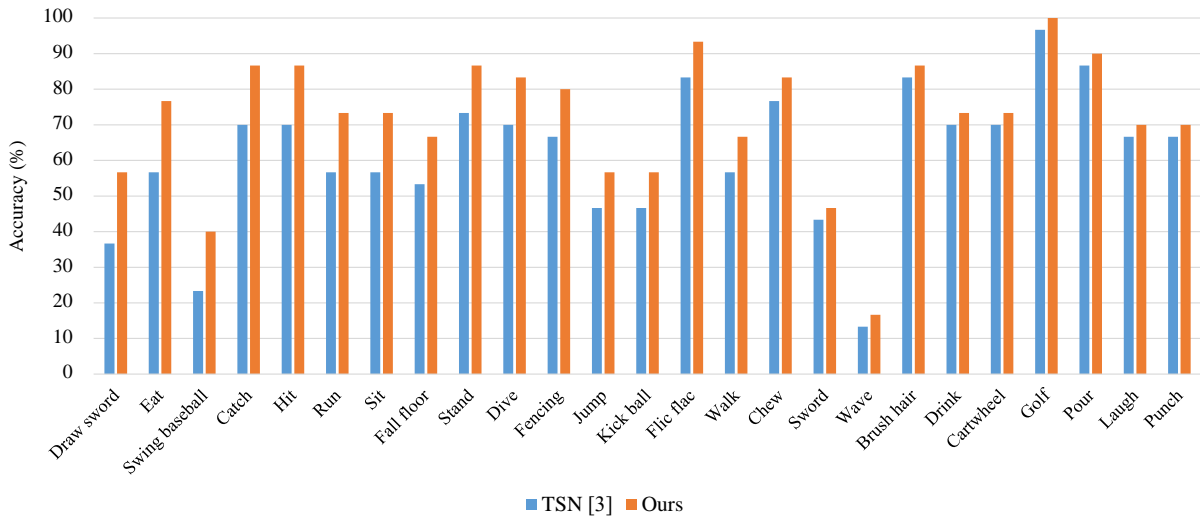


Fig. 6. Comparisons of Accuracy (%) for the top-25 classes on the HMDB51 dataset (Split 1) between our approach and the TSN model. Our approach consistently outperforms the TSN model.

TABLE II
COMPARISON ON THE NETWORK DESIGNS FOR EXPLORING TEMPORAL DYNAMICS FROM FEATURE VECTORS OF DENSELY SAMPLED FRAMES (IN ACCURACY %) ON THE HMDB51 DATASET (SPLIT 1). WE TAKE TSN [3] AS OUR BACKBONE FOR EXTRACTING FRAME LEVEL FEATURES.

Architecture	RGB	Optical Flow
TSN [3]	54.6	62.6
TSN+3-layer LSTM	51.1	59.6
TSN+TSM (AlexNet)	51.7	58.0
TSN+TSM (3-layerConvNet)	55.0	63.1

Table III shows the comparisons with the designs having fewer attention branches. “w Attn. (A_0)” denotes that only the attention A_0 is applied in the input (see Figure IV). “w Attn. (A_1 & A_2)” denotes the attentions A_1 and A_2 are applied after the first layer and the second layer of the shallow CNN. “w Attn. (A_0 & A_1 & A_2)” denotes our final scheme with all the three branches of attentions included. We can that the hierarchical attention structure provides superior performance, and when our attention model is enabled (w/ *Attn.*) with

experiments conducted on the HMDB51 dataset (Split 1), the performance can be improved by 0.8%, demonstrating the effectiveness of the attention mechanism.

Influence of the Density of Temporal Sampling. Videos of different lengths will generate *VideoMaps* of different height. We can aggregate the feature vectors of dense frames of a video to form a *VideoMap*. In practice, for the convenience of learning, we densely sample the video frames to have a fixed number of frames to form the *VideoMap* of fixed height. In the training, we set the number of frames as 256 in considering the average length of videos. In order to measure the influence of temporal frame sampling density during testing, we have compared the performance under different temporal sampling densities, *i.e.*, 25, 32, 64, 128, and 256 frames per video on the HMDB51 dataset (Split 1) and show the results in Figure 5 (a). We can see that the performance increases as the sampling density increases. This is consistent with the human perception. Our method provides a way to jointly explore the inter-frame dynamics.

In contrast, for the TSN model [3], the increase of the number of frames in the test will not increase the performance,

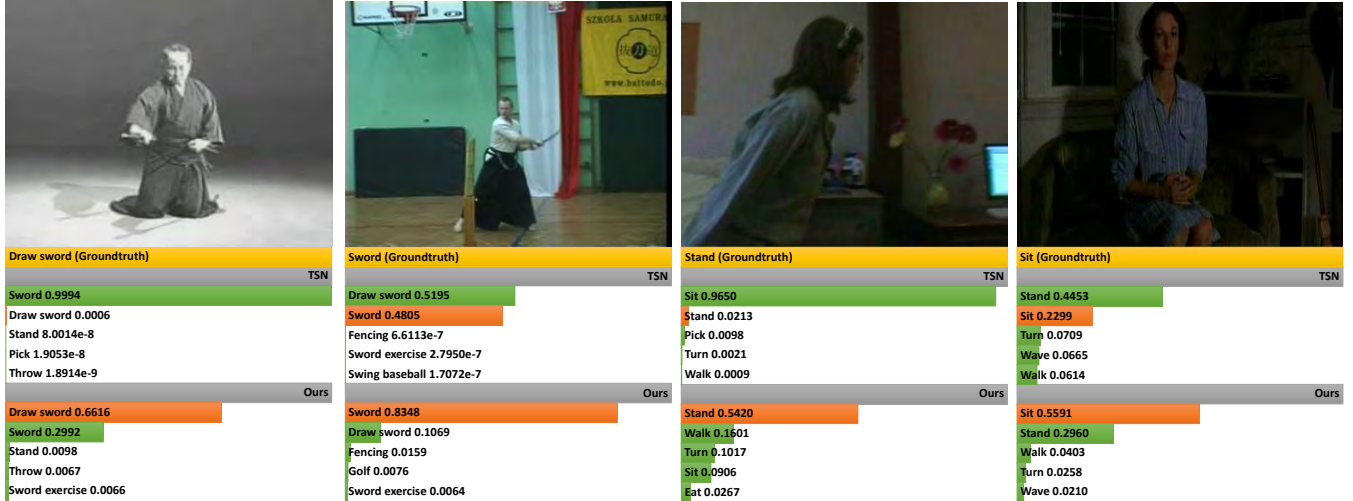


Fig. 7. Examples that our scheme succeeds in recognizing the action while TSN [3] fails. Our scheme perform well mainly due to the Temporal-Spatial Mapping operation enables the joint exploration of temporal frames and the consideration of time order.

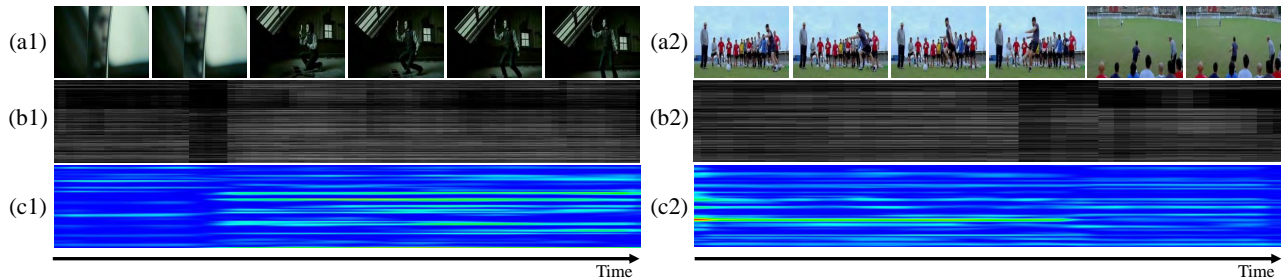


Fig. 8. Visualizations on a video of action “stand” ((a1)–(c1)) and “kick ball” ((a2)–(c2)). (a1)(a2) Video frames over time (only some frames are shown). (b1)(b2) VideoMap. (c1)(c2) Visualization from our ConvNet of the Conv-3 layer using approach Grad-CAM [40]. Note that images in (b1)(b2)(c2) are resized and horizontal axis denotes the time here. We can see that Grad-CAM map presents higher responses over temporal segments corresponding to the frames when the persons are doing the corresponding actions.

TABLE III
ACCURACY (%) OF TWO-STREAM BASED NETWORK TSN+TSM WITHOUT (W/O ATTN.) AND WITH (W/O ATTN.) TEMPORAL ATTENTION ON THE HMDB51 DATASET (SPLIT 1).

Model	RGB	Optical Flow	Fusion
w/o Attn.	55.0	63.1	72.4
w Attn. (A_0)	54.6	63.3	72.4
w Attn. (A_1 & A_2)	55.0	63.3	72.7
w Attn. (A_0 & A_1 & A_2)	55.2	63.3	73.2

and the performance saturates at 25 frames as shown in Figure 5 (b). This is because the TSN-like approaches (e.g., TSN [3], TLE [18]) explore the temporal dynamics by simply averaging/multiplying the scores/features of the frames. The statistical information rather than the per-frame detailed information is explored for recognition. Then, more frames above a certain number of frames help little when they are enough to represent the statistical information. While, the performance of our model increases with the increase of frame density (see Figure 5 (a)).

D. Comparison with the State-of-the-art

We compare our proposed scheme with the state-of-the-art approaches for video action recognition in Table IV. We evaluate the performance on the HMDB51 and the UCF101 dataset. For both datasets, we use the provided evaluation protocol and report the mean average accuracy over the three splits. We can see our scheme achieves the best performance, with 72.7% on the HMDB51 dataset and 94.3%, on the UCF101 dataset. In comparison with the TSN, we achieve 4.2% improvement on the HMDB51 dataset.

Compared with the HMDB51 dataset, the accuracy improve-

TABLE IV
PERFORMANCE COMPARISONS (IN ACCURACY %) OF OUR METHOD WITH THE OTHER STATE-OF-THE-ART METHODS OVER ALL THE THREE SPLITS.

Method	HMDB51	UCF101
Slow Fusion CNN [41]	—	65.4
Two-Stream CNN (VGG16) [5]	58.5	91.4
Two-Stream CNN (AlexNet) [2]	59.4	88.0
Key Volume Mining [4]	63.3	93.1
Two-Stream CNN Fusion [5]	65.4	92.5
Spatiotemporal ResNets [42]	66.4	93.4
TSN (BN-Inception) [3]	68.5	94.0
Spatiotemporal Multiplier Nets [43]	68.9	94.2
Spatiotemporal Pyramid Net [22]	68.9	94.6
Fusion+iDT [5]	69.2	93.5
ActionVLAD (VGG16)+iDT [28]	69.8	93.6
TLE:Bilinear [23]	71.1	95.6
LRCN [7]	—	82.9
C3D [20]	—	85.2
C3D+iDT [20]	—	90.4
C3D+LSTM [44]	55.2	85.4
VideoLSTM+iDT(FV) [9]	63.0	91.5
Multi-Granular Nets [21]	63.6	90.8
Multi-Stream Fusion [10]	—	92.6
Hierarchical Attention Nets [11]	64.3	92.7
TSN+TSM (Ours w/o Attn.)	72.2	94.1
TSN+TSM (Ours w/ Attn.)	72.7	94.3

ment on the UCF101 dataset is smaller. The performance of the UCF101 dataset is approaching saturation ($>94\%$) and it becomes difficult to demonstrate the effectiveness of an approach. We will perform further study on more challenging datasets in the future.

E. Visualization

We make performance comparison for all the categories on the HMDB51 dataset to get better insights. Figure 6 shows the top-25 classes that our approach outperforms TSN. For some action classes such as “Draw sword” and “Eat”, our scheme outperforms TSN even by 20%. In TSN, “Draw sword” is easy to be mistaken as “Sword” and “Wave” since these actions usually share some common states like waving. Figure 7 shows such an example. With our scheme capable of looking at dense frames with time order embedded rather than several sparse frames, the accuracy is improved by 23% for the class of “Draw sword”. Similarly, “Eat” and “Drink” are prone to be confused and we find the video samples of the two classes usually have frames of similar states once the cup is away from the mouth. For the classes with time order, e.g., “Stand” versus “Sit” as shown in Figure 7, our scheme can capture the time order well thanks to the *VideoMap* representation and outperforms TSN.

In addition, Figure 9 and Figure 10 show partial of the confusion matrix corresponding to some easily confused categories, as referred in Section 6.4. In Figure 9, for the approach of Temporal Segment Networks (TSN) [3], “Draw sword” is easy to be mistaken as “Sword” and “Wave” where these actions usually share some common states like waving. Our proposed scheme performs much better than TSN since it is capable of looking at all frames and jointly making decision. In Figure 10, the proposed TSN+TSM method can

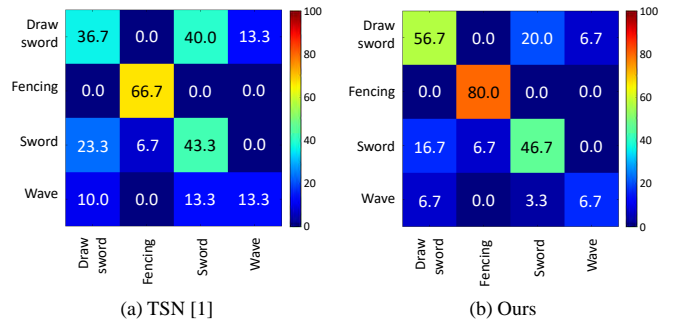


Fig. 9. Comparison of partial confusion matrix with related categories of “Draw sword”, “Fencing”, “Sword”, and “Wave”. (a) TSN [1]; (b) Ours.

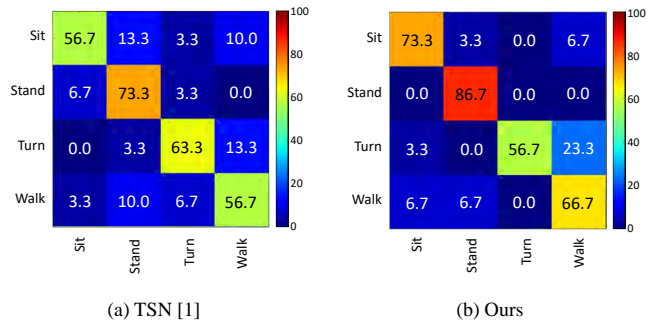


Fig. 10. Comparison of partial confusion matrix with related categories of “Sit”, “Stand”, “Turn”, and “Walk”. (a) TSN [1]; (b) Ours.

distinguish “Sit”, and “Stand” much better since the time order information is embedded in the *VideoMap* representation.

There are some failure cases of our TSM, For instance, one of action “Climb” is mistaken as “Jump”, and one of “Smoke” is mistaken as “Eat”. The video appearances in the confused classes are similar. And we show two examples in Figure 11. For instance, “Climb” is mistaken as “Jump”, and “Smoke” is mistaken as “Eat”. The appearances in the confused classes are similar.

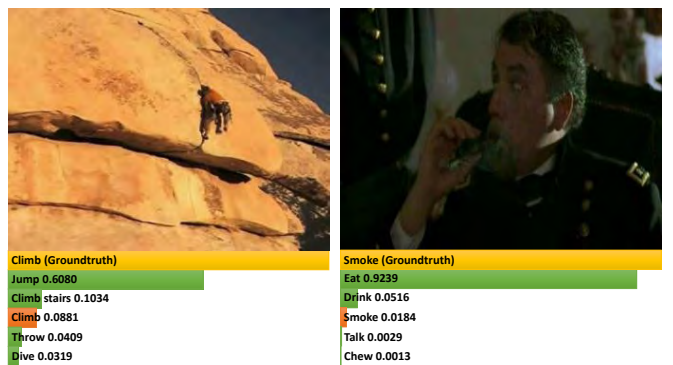


Fig. 11. Examples on the failure cases of the proposed model on the HMDB51 dataset (Split 1). The main reason for these failures comes from the similar appearances in the confusing classes.

Furthermore, we adopt Grad-CAM visualization technique [40] to analyze our head ConvNet. Grad-CAM is a class-discriminative localization technique, which can provide vi-

sual explanations from the learned ConvNet model without requiring architectural changes or re-training. In Figure 8, we can see that Grad-CAM map presents higher responses over temporal segments corresponding to the frames that persons are doing the corresponding actions. For instance, in Figure 8 (a1)–(c1), For the action of “stand up”, there are some unrelated frames before the acting of standing up. In Figure 8 (a2)–(c2), for the action “kick ball”, there are high responses at the first 2/3 of the temporal duration and lower responses at the remaining time duration. The temporal response characteristics are similar to that for object detection/classification, where the regions being highly correlated with the actions/objects/classes having higher responses.

VII. CONCLUSION

To model the temporal-spatial evolution in video for action recognition, we propose a simple yet effective operation, Temporal-Spatial Mapping (TSM), to enable the joint analysis of the dense frames of a video. We propose a video level 2D feature representation by transforming the convolutional features of a sequence to a *VideoMap*, where the temporal dynamic evolution is well embedded. We leverage a head ConvNet with temporal attention model to further explore the temporal-spatial dynamics in the VideoMap and learn effective video-level feature representation for classification. The experiment results show that the proposed scheme achieves the state-of-the-art performance, 72.7% and 94.3% on the HMDB51 and UCF101 dataset, respectively.

REFERENCES

- [1] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys*, vol. 43, no. 3, 2011.
- [2] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.
- [3] 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*, 2016, pp. 20–36.
- [4] W. Zhu, J. Hu, G. Sun, X. Cao, and 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.
- [5] C. Feichtenhofer, A. Pinz, and 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.
- [6] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and 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.
- [7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and 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.
- [8] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention,” *arXiv preprint arXiv:1511.04119*, 2015.
- [9] Z. Li, E. Gavves, M. Jain, and C. G. Snoek, “Videolstm convolves, attends and flows for action recognition,” *arXiv preprint arXiv:1607.01794*, 2016.
- [10] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, “Multi-stream multi-class fusion of deep networks for video classification,” in *Proceedings of the ACM on Multimedia Conference*, 2016, pp. 791–800.
- [11] Y. Wang, S. Wang, J. Tang, N. O’Hare, Y. Chang, and B. Li, “Hierarchical attention network for action recognition in videos,” *arXiv preprint arXiv:1607.06416*, 2016.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
- [16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European Conference on Computer Vision*, 2016, pp. 21–37.
- [17] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” *arXiv preprint arXiv:1612.08242*, 2016.
- [18] A. Diba, V. Sharma, and L. Van Gool, “Deep temporal linear encoding networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2329–2338.
- [19] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.
- [20] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.
- [21] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo, “Action recognition by learning deep multi-granular spatio-temporal video representation,” in *Proceedings of the ACM on International Conference on Multimedia Retrieval*, 2016, pp. 159–166.
- [22] Y. Wang, M. Long, J. Wang, and P. S. Yu, “Spatiotemporal pyramid network for video action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [23] A. Diba, V. Sharma, and L. Van Gool, “Deep temporal linear encoding networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016.
- [24] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.
- [25] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, “Aggregating local image descriptors into compact codes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, p. 1704, 2012.
- [26] J. Nchez, F. Perronnin, T. Mensink, and J. Verbeek, “Image classification with the fisher vector: Theory and practice,” *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.
- [27] K. Xu, X. Jiang, and T. Sun, “Two-stream dictionary learning architecture for action recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 567–576, 2017.
- [28] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, “Actionvlad: Learning spatio-temporal aggregation for action classification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3165–3174.
- [29] Z. Xu, Y. Yang, and A. G. Hauptmann, “A discriminative cnn video representation for event detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1798–1807.
- [30] X. Peng, C. Zou, Y. Qiao, and Q. Peng, “Action recognition with stacked fisher vectors,” in *European Conference on Computer Vision*, 2014, pp. 581–595.
- [31] L. Sun, K. Jia, D.-Y. Yeung, and 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.
- [32] Y. Shi, Y. Tian, Y. Wang, and T. Huang, “Sequential deep trajectory descriptor for action recognition with three-stream cnn,” *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1510–1520, 2017.
- [33] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, “Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length,” *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 634–644, 2018.
- [34] D. Li, T. Yao, L. Duan, T. Mei, and Y. Rui, “Unified spatio-temporal attention networks for action recognition in videos,” *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1–1.
- [35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

- [36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [37] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, 2015, pp. 448–456.
- [38] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563.
- [39] K. Soomro, A. R. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild,” *Computer Science*, 2012.
- [40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
- [41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and 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.
- [42] C. Feichtenhofer, A. Pinz, and R. Wildes, “Spatiotemporal residual networks for video action recognition,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3468–3476.
- [43] C. Feichtenhofer, A. Pinz, and 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.
- [44] Y. Ye and Y. Tian, “Embedding sequential information into spatiotemporal features for action recognition,” in *Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1110–1118.