Content-Attention Representation by Factorized Action-Scene Network for Action Recognition

Jingyi Hou, Xinxiao Wu, *Member, IEEE*, Yuchao Sun, and Yunde Jia, *Member, IEEE*

***Abstract*—During action recognition in videos, irrelevant motions in the background can greatly degrade the performance of recognizing specific actions with which we actually concern ourself here. In this paper, a novel deep neural network, called factorized action-scene network (FASNet), is proposed to encode and fuse the most relevant and informative semantic cues for action recognition. Specifically, we decompose the FASNet into two components. One is a newly designed encoding network, named content attention network (CANet), which encodes local spatial–temporal features to learn the action representations with good robustness to the noise of irrelevant motions. The other is a fusion network, which integrates the pretrained CANet to fuse the encoded spatial–temporal features with contextual scene feature extracted from the same video, for learning more descriptive and discriminative action representations. Moreover, different from the existing deep learning based tasks for generic action recognition, which applies softmax loss function as the training guidance, we formulate two loss functions for guiding the proposed model to accomplish more specific action recognition tasks, *i.e.*, the multilabel correlation loss for multilabel action recognition and the triplet loss for complex event detection. Extensive experiments on the Hollywood2 dataset and the TRECVID MEDTest 14 dataset show that our method achieves superior performance compared with the state-of-the-art methods.**

***Index Terms*—Deep neural network, multi-label action recognition, complex event detection.**

I. Introduction

VIDEO based action recognition is an active research area, whereas recognizing actions in realistic unconstrained videos is still quite challenging due to the factors of viewpoint variations and background clutters. As for the cluttered background, different videos may demonstrate different visual appearances of background scenes, and the background in one video may contain irrelevant motions other than the actions which we are actually interested in. Most of the existing action recognition methods lessen the impact of various visual appearances by utilizing spatio-temporal proximity information among features which are extracted from space-time interest points [1]–[3] or along motion trajectories [4], [5]. However, without the guidance of high-level semantic information, action representations extracted by these methods do not have the capability of selectively expressing the most relevant action information in a video. Taking a video captured in a street for example, there are many people walking in the street and a man getting off a car. It is obvious that a person is more likely to pay more attention to the "getting off a car" than the "walking", but machines may classify the actions in this video as the "walking".

To tackle the problem of the cluttered background for action recognition, some methods [5], [6] use the techniques of object tracking and detection before feature extraction and other methods [7]–[9] jointly recognize and localize actions in videos. But it is time-consuming, labor-intensive, and error-prone to annotate bounding boxes of persons for training. Recently, several Convolutional Neural Network (CNN) based methods are proposed to learn the most relevant representations for specified task theoretically trained in an end-to-end manner, *e.g.*, 3D CNNs [10], [11], Multi-resolution CNNs [12], and Long-term Recurrent Convolutional Networks (LRCN) [13]. These methods have achieved good performances of learning relevant representations on specific action datasets. However, they do not have strong generalization ability to get good performances on other datasets which share different kinds of action categories with their training datasets because these complex deep models are liable to be over-fitting without sufficient available labeled datasets to fine-tune these models.

In this paper, we propose a Factorized Action-Scene Network (FASNet) to eliminate irrelevant background motion information without the cumbersome action detection preprocessing or extensive labeled training data. The FASNet first factorizes action videos into action and scene components by extracting the corresponding local spatial-temporal features and static scene features, respectively. The spatial-temporal features are then encoded by a newly proposed Content Attention Network (CANet) which is a component of the FASNet, to suppress the influence of irrelevant background action for more descriptive and discriminate representations of relevant actions. The encoded relevant action representations and the useful context scene features are finally fused by the concatenate layer to obtain the action representation.

Accordingly, the training procedure of the FASNet involves two stages: (i) training the CANet, and (ii) fine-tuning the whole FASNet initialized with the pre-trained CANet. The first stage aims to make the CANet capable of automatically selecting the

Manuscript received February 14, 2017; revised April 30, 2017 and September 26, 2017; accepted October 16, 2017. Date of publication November 8, 2017; date of current version May 15, 2018. This work was supported by the Natural Science Foundation of China under Grants 61673062 and 61472038. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Benoit Huet. *(Corresponding authors: Xinxiao Wu; Yunde Jia)*

The authors are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: houjingyi@bit.edu.cn; wuxinxiao@bit.edu.cn; sunyuchao@bit.edu.cn; jiayunde@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2017.2771462

most informative and relevant local spatial-temporal features to precisely describe relevant actions. A good relevant action video representation is expected to be close to the representation of a clean and simple action video. A clean action video is defined as a video only containing a single action with relatively clean background. The CANet which encodes the local spatial-temporal features based on super vector scenario with the attention mechanism [15] is thus trained using clean action videos as the ground truth. Note that the CANet needs to be pre-trained only once via the clean video dataset. For a new action recognition task, we just integrate the CANet into the FASNet regardless of the aforementioned training guidance by the clean videos.

The second stage is to develop a network that harnesses the information of scenes and actions by combining them for action recognition. The contextual scene can improve the performance of action recognition in realistic videos because many realistic actions usually happen in some particular scenes. For example, the action of "DriveCar" often happens in an outdoor scenario. Since it is unnecessary to process scene information in each frame of a video, we just extract features of the key frames as the static scene features. After pooling and a nonlinear transformation, the static scene features are fused with the output relevant action features from the CANet by the concatenate layer.

To enhance the adaptiveness of the FASNet on some specific action recognition tasks, we employ two loss functions. One is the multi-label correlation loss which is designed for multi-label action recognition, and the other is the triplet loss for event detection. In the case of multi-label action recognition, a video could be classified as two or more action categories, *e.g.*, people might kiss while hugging. With respect to the symbiosis among different action categories, the multi-label correlation loss is used to measure the probability of the co-occurrence of action categories. In the case of complex event detection, where the training set contains a vast amount of negative video exemplars and a small amount of positive exemplars, the widely used softmax loss only considers the separable ability of features while omitting the discriminative power of features aside resulting in the poor generalization of the model. Motivated by the success of the triplet loss in retrieval tasks [16]–[19], we apply the triplet loss to event detection to train the model by maximizing the relative distance of inter-class and intra-class features through a set of triplets.

The main contributions of this paper are three folds:

1) We propose a novel deep neural network, FASNet, to encode and fuse the most relevant motion information and scene information for action recognition. To accomplish this goal, a trainable encoding network, CANet, is proposed to automatically learn the most relevant motion information from action videos.
2) We formulate two new loss functions for the training of the FASNet. The multi-label correlation loss is designed for the multi-label action recognition task and the triplet loss function is introduced for complex event detection.
3) Experiments on both Hollywood2 dataset and MEDTest 14 dataset show the superior performance of the proposed FASNet compared with the state-of-the-art methods.

II. Related Work

In this section, we discuss the previous works related to our method: covering the deep encoding networks and the fusion of action and scene for action recognition.

Deep encoding networks: Recently, deep encoding networks are prevalent for encoding local features of videos or images. The Recurrent Neural Network (RNN) based methods [20], [13] capture contextual information from video sequences. But the sheer amount of trainable parameters of RNNs makes it easy to be overfitting without large scale training data. Hierarchical rank pooling [21] stacks and rank-pools non-linear feature functions to encode the temporal information of video sequences. These approaches are only suitable for encoding temporal signals. Super vector based encoding methods [22]–[24] which map the local features of images or videos to form high dimensional representations have achieved good performances in several tasks. The most related work to our method is the NetVLAD encoding method [25] which proposes a super vector based encoding model with trainable parameters for place recognition. The main difference between the NetVLAD and the proposed CANet is that the CANet is more adaptive, content-aware and robust to the irrelevant content by addressing the attention mechanism and using clean videos as the guidance for training. Besides, the codewords of the CANet need not to be trained, thus reduces the computation cost.

Fusion of action and scene: Over the past years, combining multi-modal information, namely action and scene, has been successfully employed in action recognition [26]. Ikizler-Cinbis *et al.* [27] used a multiple instance learning framework, which integrates object, scene and action features of videos for action recognition. They employed Gist features [28] to describe the contextual scene in a video. Since the Gist features cannot express the information of objects which actually are parts of scenes, they additionally detected objects and extracted HOG features [29] from the objects. Other hand-crafted feature based works [30], [31] combine object information of scenes with action information for action recognition. These methods may lose a lot of useful information of scenes. The recent deep learning based methods overcome the limitation of global representation ability of traditional hand-crafted features on scene information. The two-stream CNNs based methods [32], [33] exploit two separated 2D CNNs, namely spatial net and temporal net, to capture motion and scene information from action videos. They pre-compute the optical flow of videos as the input of the temporal net to obtain motion information. The optical flow is generated before training, which might be not compatible with the final recognition task, and its generation procedure has high computational and storage complexities. Shi *et al.* [34] applied a three-stream CNN to capture static spatial, short-term temporal and long-term temporal information of videos for action recognition. To extract long-term motion information from videos, the Sequential Trajectory Texture images are calculated by computing the dense trajectories before training the three-stream CNN. Feichtenhofer *et al.* [35] added a 3D CNN to the two-stream CNNs, yet it is hard to train a general 3D CNN due to the insufficient large-scale labeled action video datasets. Although

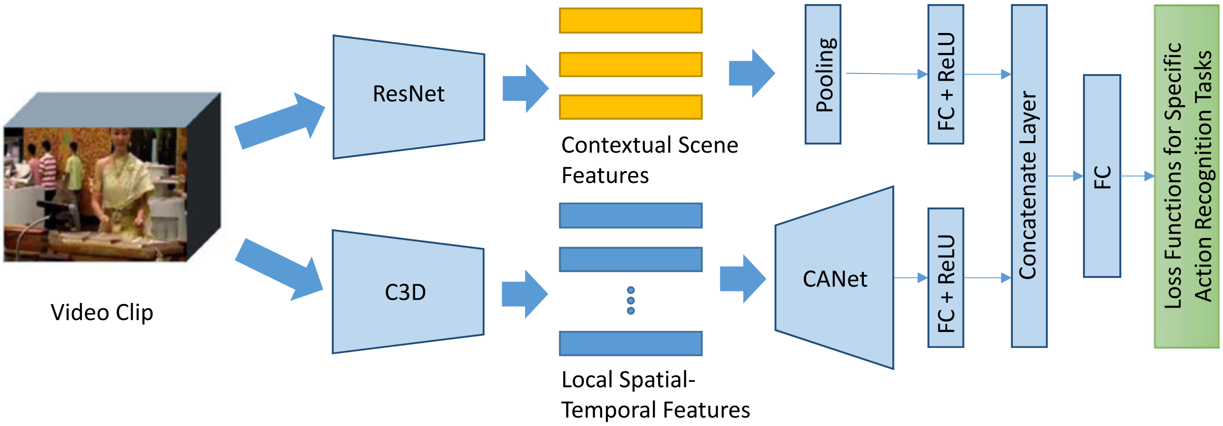

Video Clip
ResNet
C3D
Contextual Scene Features
Local Spatial-Temporal Features
Pooling
CANet
FC + ReLU
FC + ReLU
Concatenate Layer
FC
Loss Functions for Specific Action Recognition Tasks

Fig. 1. Schema of the proposed network. The FASNet factorizes action videos into action and scene components by extracting the corresponding local spatial-temporal features and static scene features with the C3D model [11] and the deep residual network (ResNet) [14], respectively. The CANet is pre-trained and added as a component of the FASNet. The loss function is designed according to different action recognition tasks.

the C3D model [11] pre-trained on sports-1M dataset shows good performance on recognizing small scale datasets, *e.g.*, the UCF101 [36] and the HMDB51 [37], where most actions are sports-related, it is impractical if we fine-tune a pre-trained CNN using a small scale dataset (*e.g.*, the Hollywood2 [38]) which shares different domains with the pre-training dataset (*e.g.*, Sports-1M). Different from these methods, our model is more adaptive by extracting local features using pre-trained 3D and 2D CNNs from raw frames of videos, and utilizes the local features as input to train a relatively simple network for a specific task.

III. Our Approach

In this section, we first adopt two existing CNNs to respectively extract local spatial-temporal features and static scene features from action videos. Then, we describe the proposed CANet to discover the most relevant cues for action description. Finally, we present the FASNet network with different novel loss functions for different action recognition tasks.

A. Feature Extraction

Action-related feature extraction: Features extracted from the bottom layers of the C3D deep model [11] are compact and full of spatial-temporal motion information, which is compatible with the Content Attention Network (CANet). Thus we utilize them as the input of the CANet. The C3D conducts all convolutions with $3 \times 3 \times 3$ 3D convolutional kernels to capture spatial and temporal information simultaneously. Here is the architecture of C3D: *Conv1a(64) - Pool1 - Conv2a(128) - Pool2 - Conv3a(256) - Conv3b(256) - Pool3 - Conv4a(512) - Conv4b(512) - Pool4 - Conv5a(512) - Conv5b(512) - Pool5 - fc6(4096) - fc7(4096) - softmax*, where *Conv1a(64)* denotes the convolutional layer with 64 filters and *fc6(4096)* is the fully connected layer with 4096 nodes. The *Pool1* to *Pool5* denote the 3D pooling layers, and all the pooling kernels of *Pool2* to *Pool5* are of size $2 \times 2 \times 2$, except for the *Pool1* whose kernel size is $1 \times 2 \times 2$. The input of C3D is a video segment with the size of $171(width) \times 128(height) \times 16(number\ of\ frames)$. Here, we use features of the *Pool5* layer for encoding due to their redundancy of local spatial and temporal information. We do not use the features extracted from the top-most fully connected layer. Since the C3D model is pre-trained with a large amount of sport-related action videos from the Sports-1M dataset, they are more likely to represent sport-related action videos. Compared to the top-most fully connected layers, the *Pool5* layer characterizes low-level appearance and motion information of actions, and the features extracted from the *Pool5* layer are thus easy to be transferred to specific tasks.

We concatenate values at the same spatial location of different feature maps as the spatial-temporal representation of a video block. As shown in Fig. 3, after *Pool5*, we have 512 feature maps, and these features can be described by a 4×4 grid of 512-d features at strided regions of a 16-frame video segment. Each 512-d vector captures rich spatial and temporal information regarding to the corresponding region of the video segment.

Scene-related feature extraction: To represent the context scene information in videos, we use the 152-layer deep residual net (ResNet)[14] to extract features of key frames from videos. The 152-layer ResNet is trained by a large-scale image dataset, *i.e.*, the ImageNet [39], and won the 1st place on ILSVRC 2015 classification task, which shows its strong generalization ability. The architecture of the ResNet is a very deep plain CNN with shortcut connections inserting into it. The structure of the plain CNN is similar to the VGG nets [40] and has even lower complexity, with 3×3 convolutional kernels and one fully connected layer. The shortcut connections perform identity mapping for fast converging to a better solution.

Since the ResNet trained on the ImageNet is generalized to describe static images, we use the 1000-d features extracted from its top-most layer to represent complex scene information of videos. The scene in an action video varies slightly, therefore no more than 3 key frames are selected from each video by finding out the top-3 frames in the video with maximum values w.r.t. frame-wise Euclidean distances.

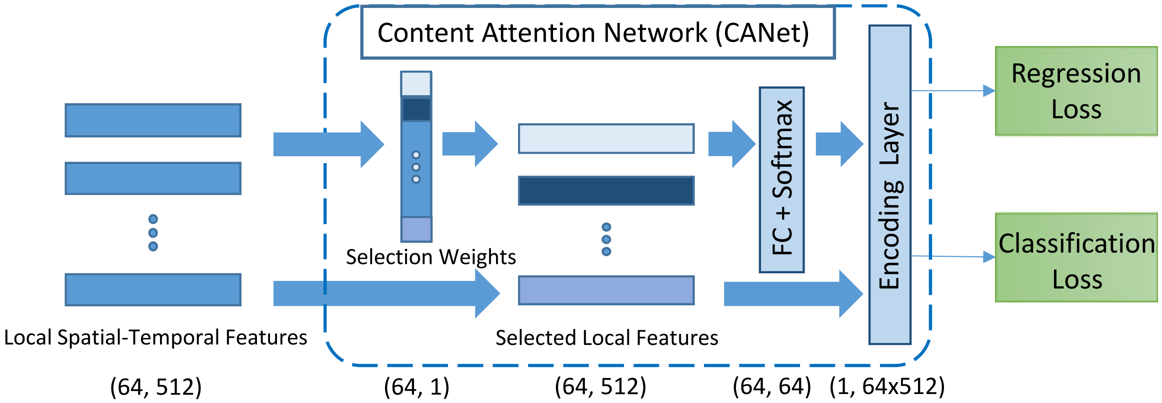


Fig. 2. The architecture of our CANet. FC denotes the fully connected layer. Inside of each bracket is the number of nodes of the corresponding layer.

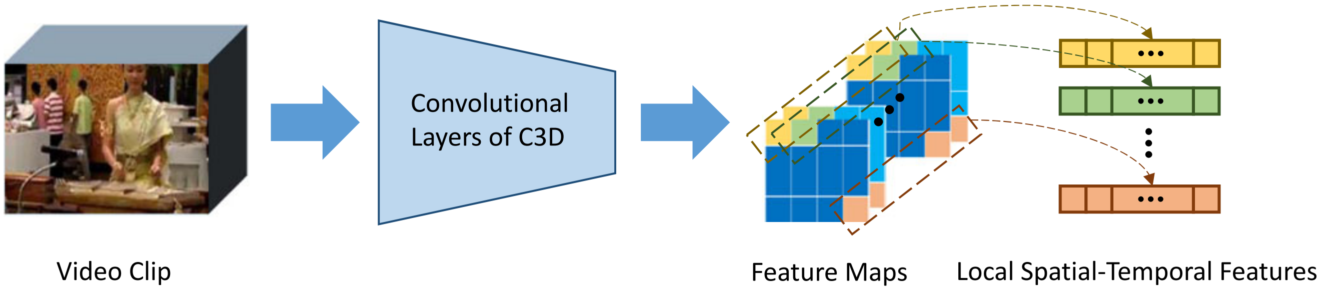


Fig. 3. Demonstration of extracting local spatial-temporal features from a video clip.

B. CANet

We train the CANet to map the local spatial-temporal features to the space of actions in clean videos and remove the irrelevant actions by the encoding process. Fig. 2 shows the architecture of the CANet. Taking the inspiration from the attention mechanism of Neural Turing Machine [15], the bottom layers of the CANet act as a selection controller to express meaningful action information and suppress irrelevant local spatial-temporal features. The fully connected layer with softmax activation is to implement the soft assignment that enables features to vote for multiple codewords and maintain more information during encoding. The soft assignment is learned by end-to-end training instead of the traditional distance measurements, so it is more reliable to achieve the global assignment. Then the features are encoded by the encoding layer in a VLAD-all [41] manner, while the codewords are generated from the local spatial-temporal features of clean action videos. Complex action videos are thus able to be represented by these clean videos to eliminate the irrelevant motion information contained in the complex videos. The regression loss is added as the auxiliary output, where the ground truth is the VLAD-all encoded features of clean videos. We first collect the clean videos covering 55 action categories for generalizing cluster centers of clean action videos from multiple public available video datasets including the KTH [42], Weizmann [43], UCF101 [36], HMDB51 [37], and the video dataset taken by us. Then, we use the C3D to obtain the local spatial-temporal features of these clean videos. Next, we cluster the local spatial-temporal features to generate 64 centers by using k-means clustering. As shown in Fig. 4, the ground-truth features of the regression loss are obtained by encoding the local spatial-temporal features of clean videos with the VLAD-all. To learn the CANet network, we use the clean action videos and the complex videos of the 55 categories in the UCF101 and the HMDB51 datasets as training data. For each training video, we segment 4 non-overlap 16-frame video clips to get totally 64 local spatial-temporal features.

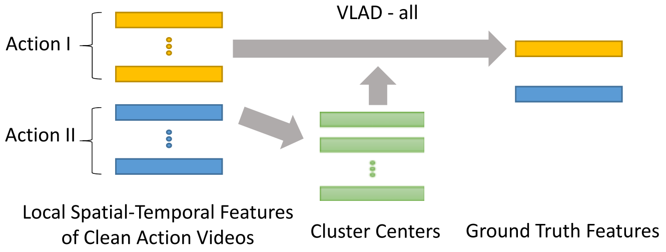


Fig. 4. Example of ground truth features of regression loss generation. Local spatial-temporal features of the same action category are encoded using VLAD-all to a vector as the ground truth of this action category.

Assume that the input local spatial-temporal features are represented by $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N] \in \mathcal{R}^{H \times N}$, where $\boldsymbol{x}_n = [x_n^1, x_n^2, ..., x_n^H]^{\mathrm{T}}$ is a H-dimensional descriptor, and N is the number of descriptors in a video. The selection weights $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_N]$ are learned from the input features $\boldsymbol{X}$ to generate the selected local features, $\boldsymbol{x}'_n = \alpha_n \boldsymbol{x}_n$.

By using a linear transformation of $\boldsymbol{X}$ with shared weights to reduce the feature dimension from H to 1, we leverage Rectified Linear Units (ReLU) to eliminate the irrelevant motion information for describing videos. Obviously, if we directly use the output of the ReLU as selected weights, the gradient will be linear w.r.t. the parameters. In order to prevent the gradient explosion problem, we use softmax activation to normalize the output of the ReLU. To ensure the values of the selected features are consistent at the order of magnitude with the codewords, we multiply the output of softmax with N. Accordingly, the selection weights are calculated.

Let $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_K] \in \mathcal{R}^{H \times K}$ denote K codewords, then the VLAD-all of the nth descriptor is calculated by

$$\boldsymbol{v}_i = \left[\omega_n^1(\boldsymbol{x}'_n - \boldsymbol{d}_1)^{\mathrm{T}}; \ldots; \omega_n^K(\boldsymbol{x}'_n - \boldsymbol{d}_K)^{\mathrm{T}}\right], \tag{1}$$

where

$$\omega_n^k = \frac{\exp(\boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{x}'_n + b_k)}{\sum_{k'=1}^{K} \exp(\boldsymbol{w}_{k'}^{\mathrm{T}} \boldsymbol{x}'_n + b'_{k'})} \tag{2}$$

is the soft assignment weight with $\boldsymbol{w}_k$ and b_k denoting the parameters of the third fully connected layer. Note that, we set the soft assignment weights trainable instead of fixed weights based on calculated distances, because the feature space is indeterminate, and it is hard to choose an appropriate type of distance. Then the encoded features are pooled together by $\boldsymbol{v} = \sum_{n=1}^{N} \boldsymbol{v}_n$, and further normalized by power and $l2$ normalization:

$$f_{\mathrm{power}}(\boldsymbol{v}^i) = \mathrm{sign}(\boldsymbol{v}^i)|\boldsymbol{v}^i|^{\frac{1}{2}}, \tag{3}$$

$$\boldsymbol{\nu} = f_{l2}(f_{\mathrm{power}}(\boldsymbol{v})) = \frac{f_{\mathrm{power}}(\boldsymbol{v})}{\|f_{\mathrm{power}}(\boldsymbol{v})\|_2}, \tag{4}$$

where $\boldsymbol{v}^i$ is the ith dimension of the encoded feature $\boldsymbol{v}$. Since it is hard to take the derivative of the function $\mathrm{sign}(\cdot)$, we use $\tanh(\cdot)$ as an approximation in the implementation.

Finally, two loss functions are added for learning video representations which are discriminative and robust to the noise of irrelevant actions. The main loss function is a hinge loss with $l2$-norm of weights of the last fully connected layer, given by

$$L_{\mathrm{main}} = \sum_{m}^{M} \left(\sum_{c}^{C} \max\left(0, 1 - y_m^c(\boldsymbol{u}_c \boldsymbol{\nu}_m + p_c)\right) + \theta \|\boldsymbol{u}_c\|_2 \right), \tag{5}$$

where $\boldsymbol{u}_c$ and p_c denote the parameters of the last fully connected layer, and $y_m^c \in \{+1, -1\}$ is the label of the mth feature of the cth class. M and C are the numbers of features and action categories, respectively. The auxiliary loss, which is a regression function calculating the cosine proximity of the feature and the ground truth feature $\boldsymbol{\nu}'$, is given by

$$L_{\mathrm{aux}} = -\sum_{m}^{M} \boldsymbol{\nu}_m^{\mathrm{T}} \boldsymbol{\nu}'_m. \tag{6}$$

Finally, the loss L_{CANet} of the CANet is formulated as

$$L_{CANet} = L_{\mathrm{main}} + \lambda L_{\mathrm{aux}}. \tag{7}$$

C. FASNet

In the previous subsection, we have designed the CANet which learns to encode meaningful action information. In this subsection, we exploit a fusion network to combine action and scene information with different loss functions for different specific action recognition tasks. Here, we do not use the softmax loss function which is most commonly used to train deep neural networks by modeling the categorical probability distribution for multi-class classification tasks. Because it would not be effective under the scenario of multi-label action recognition where a training sample can be assigned to multiple categories and the scenario of multimedia event detection where the quantity of training samples in each category is unbalanced or not adequate. Therefore, we present a new loss function named *multi-label correlation loss* for multi-label action recognition by exploring the symbiosis relationship among different actions. For multimedia event detection, we adopt the triplet loss [16], [17] which is a weakly supervised signal to address the problem of unbalanced number of training data in different categories.

Multi-Label Correlation Loss: In many scenarios, people perform different kinds of actions almost simultaneously. Apparently, there are some correlations among these actions in terms of their occurrence probabilities. Therefore, we propose a multi-label correlation loss to take the relationships among actions into consideration. Let $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_N] \in \mathcal{R}^{C \times N}$ be the output of the last fully connected layer, where $\boldsymbol{s}_n = [s_n^1, s_n^2, ..., s_n^C]^{\mathrm{T}}$ with C being the number of action categories. The multi-label correlation loss is formulated as

$$\begin{aligned} L_{\mathrm{corr}} = & -\sum_{n}\sum_{c} y_n^c \log s_n^c + (1 - y_n^c)\log(1 - s_n^c) \\ & -\sum_{n}\sum_{i \neq j} \gamma_{ij} D_{KL}(s_n^i || s_n^j), \end{aligned} \tag{8}$$

where $y_n^c \in \{1, 0\}$ is the cth class label of the nth video. $D_{KL}(s_n^i||s_n^j) = s_n^i \log \frac{s_n^i}{s_n^j} + (1 - s_n^i)\log \frac{1 - s_n^i}{1 - s_n^j}$ is the Kullback-Leibler divergence which aims to measure the difference of the co-occurrence probability between s_n^i and s_n^j. γ_{ij} is a factor reflecting that to what extent the ith and jth classes co-occur. To reduce the complexity of our model, we manually set each co-occurrence factor γ_{ij} empirically according to the percent of the co-occurred action videos to the total number of action videos of both the ith and ith classes. Details of setting the parameter γ will be discussed in Section IV. To ensure $s_n^c \in (0, 1)$, we add sigmoid activation to the last fully connected layer. The gradient of the loss L_{corr} with respect to s_n^c is calculated by

$$\begin{aligned} \frac{\partial L_{\mathrm{corr}}}{\partial s_n^c} = & \frac{s_n^c - y_n^c}{s_n^c(1 - s_n^c)} \\ & + \sum_{c \neq i} \left(\gamma_{ci} \log \frac{s_n^c(1 - s_n^i)}{s_n^i(1 - s_n^c)} + \gamma_{ic} \frac{s_n^c - s_n^i}{s_n^c(1 - s_n^c)} \right). \end{aligned} \tag{9}$$

The $l1$-norm is applied to each output vector s_n to enforce its sparsity, because people cannot take many actions at the same time.

Triplet Loss: Technically, tasks for multimedia event detection are more like the retrieval tasks, where each category has only a few positive exemplars. For instance, there are 8,030 training videos identified to 20 categories in the MEDTest 14 dataset which is the largest publicly available video corpora for event detection, but each event class contains only about 100 positive videos in the 100Ex scenario. The triplet loss is widely applied to person re-identification [16], [19], face recognition [17], [44], [45], and object retrieval [18]. It aims to verify identity by comparing descriptors in Euclidean space. Specifically, the input of the triplet loss layer is a set of the triplet units, $\{(\boldsymbol{t}_i^a, \boldsymbol{t}_i^p, \boldsymbol{t}_i^n)\} \subseteq \mathcal{T}$, where $\boldsymbol{t}_i^a$, $\boldsymbol{t}_i^p$ and $\boldsymbol{t}_i^n$ denote the anchor, positive and negative descriptors of the ith output of the last fully connected layer, respectively. $\mathcal{T}$ is the set of all possible triplets in the training set and has the cardinality N. We hope that $\boldsymbol{t}_i^a$ shares the same identity with $\boldsymbol{t}_i^p$ and is different from $\boldsymbol{t}_i^n$ as much as possible. Thus the constraint is set to

$$\|\boldsymbol{t}_i^a - \boldsymbol{t}_i^p\|^2 < \|\boldsymbol{t}_i^a - \boldsymbol{t}_i^n\|^2, \|\boldsymbol{t}\|_2^2 = 1, \qquad (10)$$

which means that the anchor descriptor is closer to the positive descriptor than the negative descriptor in Euclidean space with the margin of β. Consequently, the triplet loss is defined as

$$L_{\text{triplet}} = \sum_{i}^{N} \max(\|\boldsymbol{t}_i^a - \boldsymbol{t}_i^p\|^2 - \|\boldsymbol{t}_i^a - \boldsymbol{t}_i^n\|^2 + \beta, 0). \quad (11)$$

It is infeasible to accumulate all the possible triplets over the whole training set for L_{triplet} due to the high computation cost. In order to ensure fast convergence and good optimization, we select a part of triplets including hard negative triplets and random triplets. For a batch of K samples, we enforce that the batch can be divided into $K/2$ pairs where each pair contains two samples with the same label, *i.e.*, $\boldsymbol{t}^a$ and $\boldsymbol{t}^p$. Samples with different labels can be combined with $\boldsymbol{t}^a$ and $\boldsymbol{t}^p$ to form triples. For $\boldsymbol{t}^a$, M triplets are selected where the hard negative triplets take the percentage of η $(0 \leq \eta \leq 1)$ which can be tuned in real applications. In other words, the number of hard negative triplets is ηM and the number of random triplets is $(1 - \eta)M$. The hard negative triplets for $\boldsymbol{t}^a$ are generated by selecting the ηM farthest samples to $\boldsymbol{t}^a$, and the random triplets are generated by selecting samples of other classes randomly except for the samples selected in hard negative triplets. For $\boldsymbol{t}^q$, M triplets can also be selected in the same way. Therefore, we have $N = K/2 \times 2 \times M = KM$ triples for a batch of K samples.

Training protocol of the FASNet: For a new action recognition task, the FASNet (see Fig. 1) described before needs to be trained by a given dataset. Firstly, we extract the scene features and local spatial-temporal features of the videos in the dataset using the ImageNet pre-trained ResNet and Sports-1M pre-trained C3D, respectively. The scene features of each video are aggregated into one descriptor by the average pooling operation. Secondly, the CANet pre-trained using the clean action video dataset and the complex videos of 55 categories in the UCF101 and the HMDB51 is fine-tuned by using the local spatial-temporal features of the given training dataset. We remove the last fully connected layer of the CANet, and make it to be a part of the FASNet. Thirdly, the FASNet is trained with the loss for the specific task by fixing the weights of the CANet part until the FASNet converges. Finally, we set the part of the CANet to be trainable, and fine-tune the entire FASNet.

IV. Experiments and Results

A. Datasets

Extensive experiments are conducted on the Hollywood2 [38] and the TRECVID MEDTest 14[1] datasets to evaluate the performance of our method. Mean Average Precision (mAP) is applied to evaluate the performance of the proposed method on both Hollywood2 and TRECVD MEDTest 14 datasets.

The Hollywood2 dataset contains 12 action categories, including “AnswerPhone”, “DriveCar”, “Eat”, “FightPerson”, “GetOutCar”, “HandShake”, “HugPerson”, “Kiss”, “Run”, “SitDown”, “SitUp” and “StandUp” with 3,669 video clips which are collected from 69 different Hollywood movies. In our experiments, we use the video dataset with 1,707 action videos and split it into a training set of 823 videos and a test set of 844 videos by following the standard split strategy as [38]. Different from other action datasets, such as the UCF101 [36] and the HMDB51 [37] datasets, a video of the Hollywood2 dataset can be classified to multiple classes. Accordingly, on this dataset, we fine-tune our model by using the multi-label correlation loss.

The TRECVID MEDTest 14 dataset consists of 20 event categories which are identified as E21-E40, namely “attempting a bike trick”, “cleaning an appliance”, “dog show”, “giving directions to a location”, “marriage proposal”, “renovating a home”, “rock climbing”, “town hall meeting”, “winning a race without a vehicle”, “working on a metal crafts project”, “beekeeping”, “wedding shower”, “non-motorized vehicle repair”, “fixing musical instrument”, “horse riding competition”, “felling a tree”, “parking a vehicle”, “playing fetch”, “tailgating” and “tuning musical instrument”. We conduct our experiments in the EK100 scenario where each event class has about 100 positive training videos. There are totally 8,030 training videos with 4,983 negative exemplars. The testing dataset has about 23,000 videos. As far as we know, the MEDTest 14 dataset is the largest publicly available video corpora for event detection. Since actions are the major constituent parts of almost all the events, we consider event detection tasks to be a kind of application of action recognition tasks. Apparently, classifiers trained on the MEDTest 14 dataset are prone to overfit due to the unbalanced quantity of positive and negative exemplars. Therefore, we use the triplet loss for this task.

The clean dataset consists of the KTH dataset, the Weizmann dataset as well as a part of the UCF101 and HMDB51 datasets. The KTH and the Weizmann are constrained action datasets, and the backgrounds of videos in these datasets are simple without any irrelevant action. Therefore, we take these two entire datasets into our clean video dataset. The UCF101 and HMDB51 are realistic action datasets, of which videos are mainly obtained from movies and the Internet. We just select videos with relevant actions and simple backgrounds to form the clean action dataset. Here, background is supposed to be simple, when background objects are relative still or with subtle movements to the ground. Some categories of the clean videos collected from these 4 publicly available datasets still have

[1] http://www.nist.gov/itl/iad/mig/med14.cfm.

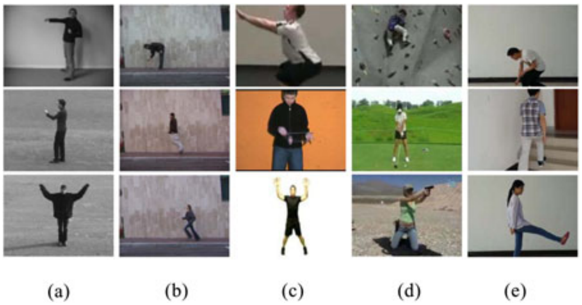


Fig. 5. Example frames of the clean video dataset. Frames in the same column are from the same dataset. (a) KTH, (b) Weizmann, (c) UCF101, (d) HMDB51, and (e) Self-collected dataset.

insufficient number of exemplars, so we record some action videos by ourselves as supplementary. There are 137 video clips performed by 5 subjects taken by ourselves. Totally, there are 780 clean action videos with 55 clean categories. Fig. 5 shows the example frames of the clean videos.

The pre-training videos of the CANet are the clean videos and the complex videos with the 55 categories in the UCF101 and HMDB51 datasets with totally 23,996 videos. Although it seems that the procedures of collection, annotation and feature extraction of these clean videos are labor-intensive and time-consuming, it is worth noting that once the CANet is trained, there is no need to collect new clean videos to fine-tune the CANet for new action recognition tasks. The ground truth of the regression loss and codewords which have been calculated from the clean videos are no longer changed.

B. Implementation Details

Preprocessing: The size of each input video of the C3D model pre-trained on the Sports-1M is 16 × 128 × 171. To obtain more and finer information, we take 4 groups of non-overlap 16-consecutive frame video clips from each video. In fact, more information from each video as input is better for describing the video. But due to the limitation of computation resources, we have to fix the input length of the CANet. For the videos which are used for pre-training the CANet, we just choose videos which contain more than 63 frames. If a video has more than 64 frames, we uniformly select 4 groups of 16 consecutive frames. We resize each frame by making its shorter size equal to 171 pixels and preserve the aspect ratio of the frames. Afterwards, the 171 × 128 crop is sampled from the center of each frame.

For the videos of Hollywood2 dataset and MEDTest 14 dataset, if a video contains less than 64 frames, we pad the ends of video with its first and last frames. If a video has more than 64 frames, we uniformly select 4 groups of 16 consecutive frames. We resize each frame with shorter edge to 171 while keeping its shape unchanged. Unlike the cropping method mentioned above, we first crop the resized frame to a 171 × 171 crop. Then, for data augmentation, 6 crops are randomly sampled from each frame of the video and its horizontal flip.

We use Theano [46] toolkit for the experiments of the neural networks designed by us and C3D [11] program for feature extraction on NVIDIA GeForce GTX TITAN X GPU with 12 GB memory.

Features for fusion: Besides the local spatial-temporal features and scene-related features, we also apply Multi-skIp Feature Stacking (MIFS) [47] to the action recognition task of the Hollywood2 dataset to obtain more sufficient representations of the videos. To reduce the complexity of CANet, PCA is applied to reduce the dimensionality of MIFS into 1,600 with more than 90% information preserved.

Parameters: For the loss of CANet, the parameters θ and λ are set to 0.02 and 500,000, respectively. For the parameters γ of the multi-label correlation loss, we set $\gamma_{ij} = \gamma_{ji}$. After analyzing the co-occurrence probability of actions in the Hollywood2 dataset, we empirically set correlation parameters γ of the "HugPerson" and the "Kiss" to 0.009, the "FightPerson" and the "run" to 0.0005, the "SitDown" and the "StandUp" to 0.0005, and others to 0 (*i.e.*, the classes are supposed to be irrelevant to each other). We test the result with litter change in this parameter setting, and find that the result changes little, but it changes much when this parameter is set on other orders of magnitude, such as 0.09 and 0.9. The probable reason is that this parameter reflects the relative importance of the two terms in the multi-label correlation loss shown in Eq. 8. For the triplet loss, we set the parameter β to 1.

C. Performance of the CANet

Intuitively, the proposed CANet aims to learn more discriminative and meaningful representations of action videos. In order to evaluate the quality of the proposed CANet, we use the video representations learned by the CANet and linear SVM classifiers for action recognition (CANet+SVM). Additionally, we directly use the prediction scores of the CANet for recognition (CANet Score). We also compare the proposed methods (CANet+SVM and CANet Score) with two baseline methods as follows:

1) *C3D:* Following [11], we extract C3D fully-connected layer features from action videos with 4 temporal strides, and use linear SVM classifiers for action recognition.
2) *C3D+VLAD:* We encode the local spatial-temporal features by VLAD-all with 64 centers, and use linear SVM classifiers for action recognition.

Fig. 6 shows the per-action AP comparison among these methods on the Hollywood2 dataset. Our method outperforms others in most action classes. We notice that performances on videos containing cluttered background improve a lot, such as videos in "AnswerPhone", "GetOutCar", *etc*. Performances on videos containing only the close-up of actions improve a little, such as videos in "DriveCar" and "Kiss". It indicates that the proposed CANet can eliminate irrelevant motion information and preserve more meaningful information. The probable reason of the inferior performance of the C3D is that it is pre-trained on a sports-related action dataset and the fully-connected layer may learn the representations of the sports related actions. The overall better performance of the proposed CANet than VLAD-all shows that features encoded by the CANet can unearth more meaningful information for action recognition.

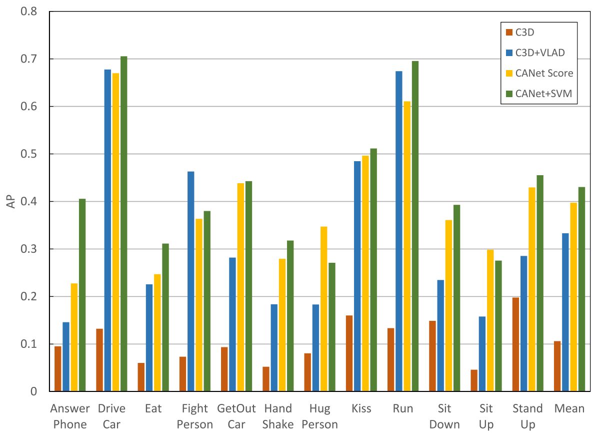


Fig. 6. Average precisions of the 12 classes of videos on the Hollywood2 dataset.

D. Performance of the FASNet

An appropriate feature fusion method may significantly enhance the performance of action recognition. Here we investigate the effectiveness and efficiency of the proposed FASNet on the Hollywood2 and the MEDTest 14 datasets.

On the Hollywood2 dataset, action recognition is performed by linear SVM classifiers (FASNet+SVM) and the output prediction scores of the FASNet (FASNet Score). On the MEDTest 14 dataset, we extract action-scene features learned by the FASNet and apply linear SVM classifiers for complex event detection (FASNet+SVM). Then, we compare the proposed fusion methods (FASNet+SVM and FASNet Score) with two baseline methods as follows:

1) *CANet+SVM and ResNet+SVM:* To demonstrate that fusing multiple features is beneficial to data representation, we extract features from CANet and ResNet, respectively, and apply linear SVM classifiers for action recognition and event detection.
2) *CANet+ResNet+GMKL:* To evaluate the efficiency of our fusion strategy, we apply the widely used feature fusion method of Generalized Multiple Kernel Learning (GMKL) [48] with linear kernels to fuse the features of the CANet and the ResNet for action recognition and event detection.

Table I presents the comparison of our fusion method using FASNet with the baseline methods. All the results show that the FASNet based methods are better than both the single feature based methods and the multi-feature fusion based methods on the Hollywood2 and the MEDTest 14 datasets. On the MEDTest 14 dataset which contains complex events, motion information is so noisy that it somehow degrades the performance, resulting in the lower mAP of the CANet than the ResNet. However, by fusing the motion and scene features, the performance is significantly improved, which suggests the complementarity of the two kinds of features.

E. Comparison With the State-of-the-Art Methods

To show the feasibility of our method for action recognition, we compare our method with the state-of-the-art methods. For the Hollywood2 dataset, we fuse the MIFS features as complementary information with the features of the proposed FASNet (FASNet+MIFS+SVM), and compare our methods with several state-of-the-art methods [5], [21], [23], [47], [49]–[52]. Sun *et al.* [49] developed a Deeply-Learned Slow Feature Analysis (DL-SFA) structure which contains one convolutional layer and two pooling layers to learn abstract and robust features with the guidance of SFA for action recognition. Jain *et al.* [50] encoded the output features of the softmax layer of CNN for object classification to recognize actions in videos. They also combined the deep features with IT features, which leads to satisfactory results. Sharma *et al.* [51] presented recurrent soft attention based models for action recognition to automatically select the important elements in video frames. Fernando *et al.* [21] proposed a hierarchical rank pooling method which uses non-linear rank pooling to aggregate frame-based deep CNN features. They also combined their hierarchical rank pooled features with encoded IT features using average kernel method, which yields better action recognition performances. The IT features proposed by Wang *et al.* [5] are introduced above, and here the features are encoded by using the Fisher Vectors (FV) method [23]. Lan *et al.* [47] developed the MIFS method as auxiliary features for action recognition on the Hollywood2 dataset. Fernando *et al.* [52] proposed an unsupervised rank pooling method to model the evolution on motion information in videos.

TABLE I
ACTION RECOGNITION PERFORMANCE (MAP, %) WITH AND WITHOUT FEATURE FUSION METHODS ON THE HOLLYWOOD2 DATASET AND MEDTEST 14 DATASET

Method	Hollywood2	MEDTest 14
Using single feature		
CANet+SVM	43.0	30.6
ResNet+SVM	32.9	34.2
Fusing multiple features with GMKL		
CANet+ResNet+GMKL	50.6	38.7
Our methods		
FASNet Score	55.5	-
FASNet+SVM	59.9	41.0

From Table II, we can observe that approaches using deep features perform much poorer than that using hand-crafted features. It indicates that the video representations of deep networks is not general enough due to the lack of labeled videos. Since 2D CNNs pre-trained on the ImageNet dataset are general enough on image-based tasks, they can also sufficiently represent the scene information of each frame in a video. However, the important motion information of actions is discarded. 3D CNNs pre-trained on the Sports-1M dataset can capture motion information yet still not general enough w.r.t. high level semantics. Our approach fusing deep 2D and 3D CNN features performs well among other deep feature based approaches, which shows the effectiveness of our approach by capturing both the scene and motion information. Moreover, our method of fusing the MIFS and deep features outperforms other state-of-the-art methods, which indicates that hand-crafted features MIFS are

TABLE II
ACTION RECOGNITION RESULTS OF DIFFERENT METHODS ON THE HOLLYWOOD2 DATASET

Method	3D/2D CNN	mAP (%)
Using hand-crafted features only		
Wang *et al.* [5]	-	64.3
Lan *et al.* [47]	-	68.0
Fernando *et al.* [52]	-	73.7
Using deep features only		
Sun *et al.* [49]	3D	48.1
Jain *et al.* [50]	2D	38.4
Sharma *et al.* [51]	2D	43.9
Fernando *et al.* [21]	2D	56.8
FASNet+SVM	3D+2D	59.9
Fusing deep features and hand-crafted features		
Jain *et al.* [50]	2D	66.6
Fernando *et al.* [21]	2D	76.7
FASNet+MIFS+SVM	3D+2D	78.1

TABLE III
EVENT DETECTION RESULTS OF DIFFERENT METHODS ON THE MEDTEST 14 DATASET

Method	3D/2D CNN	mAP (%)
Using hand-crafted features		
Wang *et al.* [5]	-	27.0
Lan *et al.* [47]	-	29.0
Fusing deep features and hand-crafted features		
Zha *et al.* [54]	2D	38.7
Using deep features		
Xu *et al.* [53]	2D	36.8
FASNet+SVM	3D+2D	41.0

complementary to the deep features from the CANet on the Hollywood2 dataset. The probable reason is that the MIFS encodes the improved trajectory (IT) features which represent the low-level gradients along optical flows while CANet encodes the higher level C3D features which contain more abstract semantics.

For the MEDTest 14 dataset, we compare the proposed method with several state-of-the-art methods of [5], [47], [53], [54] Xu *et al.* [53] proposed to use frame-level CNN features and Latent Concept Descriptors (LCD) encoded by VLAD, and combined the features with IT features for event detection. Zha *et al.* [54] combined IT features with spatial-temporal pooling of frame-level CNN features for event detection. As shown in Table III, our method outperforms all the state-of-the-art methods on the MEDTest 14 dataset by taking advantages of action and scene information without using hand-craft features. Unlike action recognition tasks on the Hollywood2 dataset, event detection performances of hand-crafted feature based methods on the MEDTest 14 dataset are even worse than 2D CNN based methods. It is probably because videos in the MEDTest 14 dataset are more complex and contain more irrelevant motion information than videos in the Hollywood2 dataset. Meanwhile, scene information is more discriminative than simply encoding the noisy local motion information extracted from the videos for complex event detection. Since we fuse scene features extracted from 2D CNN and the most relevant action features learned from the proposed CANet, our method achieves the best performance.

In this paper, we focus on action recognition tasks. The propose method can only be used in human action related tasks so far, since the clean video dataset only contains human actions. There are various motions performed by different objects in the real world. For wider range of of applications, such as the CDnet challenge [55], the mechanism of the proposed method could be used by extending the clean video dataset with motions of different kinds of objects.

V. CONCLUSION

We have proposed a novel deep learning model, Factorized Action-Scene Network (FASNet), by integrating the Content Attention Network (CANet) to fuse the most relevant motion information and useful scene information for action recognition. We also have formulated two loss functions, namely multi-label correlation loss and triplet loss, as guidance of the proposed deep architecture to learn more suitable representations for specific complex action recognition tasks. Empirical evaluations on different datasets have demonstrated that the proposed FASNet can effectively exploit more descriptive and discriminative motion and scene information from realistic videos, and is thus feasible to recognize actions in complex videos. In the future, we plan to develop a new algorithm to find key frames from video and learn more representative and compact local spatial-temporal features for the FASNet.

REFERENCES

[1] I. Laptev, “On space-time interest points,” *Int. J. Comput. Vision*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[2] J. C. Niebles, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words,” *Int. J. Comput. Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[3] S. Samanta and B. Chanda, “Space-time facet model for human activity classification,” *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1525–1535, Oct. 2014.

[4] J. Sun *et al.*, “Hierarchical spatio-temporal context modeling for action recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2009, pp. 2004–2011.

[5] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proc. IEEE Conf. Comput. Vision*, 2013, pp. 3551–3558.

[6] M. Raptis, I. Kokkinos, and S. Soatto, “Discovering discriminative action parts from mid-level video representations,” in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2012, pp. 1242–1249.

[7] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff, “Action recognition and localization by hierarchical space-time segments,” in *Proc. IEEE Conf. Comput. Vision*, 2013, pp. 2744–2751.

[8] C. Liu, X. Wu, and Y. Jia, “Transfer latent SVM for joint recognition and localization of actions in videos,” *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2596–2608, Nov. 2016.

[9] Z. Zhou, F. Shi, and W. Wu, “Learning spatial and temporal extents of human actions for action detection,” *IEEE Trans. Multimedia*, vol. 17, no. 4, pp. 512–525, Apr. 2015.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *Proc. IEEE Conf. Comput. Vision*, 2015, pp. 4489–4497.

[12] A. Karpathy *et al.*, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2014, pp. 1725–1732.

[13] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 2625–2634.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2016, pp. 770–778.

[15] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *CoRR*, vol. abs/1410.5401, 2014.

[16] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recog.*, vol. 48, no. 10, pp. 2993–3003, 2015.

[17] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 815–823.

[18] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2016, pp. 2167–2175.

[19] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2016, pp. 1335–1344.

[20] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Conf. Comput. Vision*, 2015, pp. 4041–4049.

[21] B. Fernando, P. Anderson, M. Hutter, and S. Gould, "Discriminative hierarchical rank pooling for activity recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2016, pp. 1924–1932.

[22] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 141–154.

[23] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 143–156.

[24] H. Jegou *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[25] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2016, pp. 5297–5307.

[26] K. Wang, Q. Yin, W. Wang, S. Wu, and L. Wang, "A comprehensive survey on cross-modal retrieval," *arXiv:1607.06215*.

[27] N. Ikizler-Cinbis, and S. Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 494–507.

[28] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, vol. 1, 2005, pp. 886–893.

[30] D. Han, L. Bo, and C. Sminchisescu, "Selection and context for action recognition," in *Proc. IEEE Conf. Comput. Vision*, vol. 9, 2009, pp. 1933–1940.

[31] D. J. Moore, I. A. Essa, and M. H. Hayes, "Exploiting human actions and object context for recognition tasks," in *Proc. IEEE Conf. Comput. Vision*, vol. 1, 1999, pp. 80–86.

[32] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[33] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 20–36.

[34] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream CNN," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1510–1520, Jul. 2017.

[35] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf.Comput. Vision Pattern Recogn.*, 2016, pp. 1933–1941.

[36] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv:1212.0402*.

[37] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Conf. Comput. Vision*, 2011, pp. 2556–2563.

[38] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2009, pp. 2929–2936.

[39] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[41] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Comput. Vision Image Understanding*, vol. 150, pp. 109–125, 2016.

[42] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Space-time interest points," in *Proc. IEEE Conf. Comput. Vision*, 2011, pp. 2556–2563.

[43] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[44] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vision Conf.*, vol. 1, no. 3, 2015, p. 6.

[45] Z. Dong, S. Jia, T. Wu, and M. Pei, "Face video retrieval via deep learning of binary hash representations," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3471–3477.

[46] F. Bastien *et al.* "Theano: New features and speed improvements," in *Proc. Deep Learn. Unsupervised Feature Learn. NIPS Workshop*, 2012, 2012, pp. 1–10.

[47] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 204–212.

[48] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 1065–1072.

[49] L. Sun *et al.*, "DL-SFA: Deeply-learned slow feature analysis for action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2014, pp. 2625–2632.

[50] M. Jain, J. C. van Gemert, and C. G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?" in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 46–55.

[51] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv:1511.04119*, pp. 1–11, 2016.

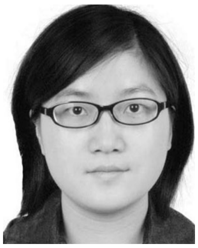
[52] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 5378–5387.

[53] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 1798–1807.

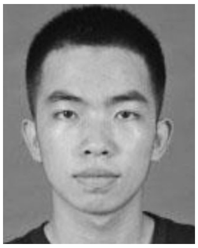
[54] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification," in *Proc. Brit. Mach. Vision Conf.*, 2015, pp. 60.1–60.13.

[55] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. 2012 IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog. Workshops*, 2012, pp. 1–8.

Jingyi Hou received the B.S. degree in electrical engineering and automation from the China University of Mining and Technology, Beijing, China, in 2014. She is currently working toward the Ph.D. degree in the Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Her research interests include computer vision, pattern recognition, and video content analysis.



Xinxiao Wu received the B.S. degree from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2010. She is currently an Associate Professor with the Beijing Institute of Technology. Her research interests include computer vision, machine learning, and video content analysis.



Yuchao Sun received the B.S. degree in 2016 from the Beijing Institute of Technology, Beijing, China, where he is currently working toward the M.S. degree in the Department of Computer Science and Technology.

Yunde Jia received the B.S., M.S., and Ph.D. degrees in mechatronics from the Beijing Institute of Technology (BIT), Beijing, China, in 1983, 1986, and 2000, respectively. He is currently a Professor of computer science with BIT, and serves as the Director of the Beijing Laboratory of Intelligent Information Technology. He has previously served as the Executive Dean of the School of Computer Science, BIT from 2005 to 2008. He was a Visiting Scientist with Carnegie Mellon University from 1995 to 1997, and a Visiting Fellow with Australian National University, in 2011. His current research interests include computer vision, media computing, and intelligent systems.