Multiscale Deep Alternative Neural Network for Large-Scale Video Classification

Jinzhuo Wang, Wenmin Wang, *Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*—With the rapid increase in the amount of multimedia data, video classification has become a demanding and challenging research topic. Compared with image classification, video classification requires mapping a video that contains hundreds of frames to semantic tags, which poses many challenges to the direct use of advanced models originally designed for image-oriented tasks. On the other hand, continuous frames in a video also give us more visual clues that we can leverage to achieve better classification. One of the most important clues is the context in the spatiotemporal domain. In this paper, we introduce the multiscale deep alternative neural network (DANN), a novel architecture combining the strengths of both convolutional neural network and recurrent neural networks to achieve a deep network that can collect rich context hierarchies for video classification. In particular, the DANN is stacked with alternative layers, each of which consists of a volumetric convolutional layer followed by a recurrent layer. The former acts as a local feature learner, whereas the latter is used to collect contexts. Compared with popular deep feed-forward neural networks, the DANN learns local features and their contexts from the very beginning. This setting enables preserving context evolutions, which we show to be essential for improving the accuracy of video classification. To release the full potential of the DANN, we develop a deeper version with stochastic-layer skip-connections and construct a multiscale DANN to incorporate contexts at different scales. We show how to apply the multiscale DANN for video classification with carefully designed configurations in terms of both input–output settings and training–testing methods. The DANN is shown to be robust to not only human-centric videos, but also natural videos. As there are few large-scale natural disaster video datasets, we construct a new large-scale one and make it publicly available. Experiments on four datasets show the effectiveness of our method for both human actions and natural events.**

***Index Terms*—Video classification, deep alternative neural network, multi-scale deep network, human activity, natural disaster, new dataset.**

I. Introduction

The sheer volume of video data currently available requires the development of robust video classification techniques that can effectively make semantic predictions for applications such as video search, summarization, intelligent surveillance, human computer interaction and multimedia information management [1], [2]. The variety of realistic video data results in many challenges for video classification, such as large intra-class variations, noisy contents unrelated to the video topic, and complex temporal structures [3], [4]. Most existing studies focus on human activities [5]–[7], and lack the recognition ability for natural videos such as those of extreme events. This paper contributes a new natural disaster video dataset and aims at proposing robust solutions for general video classification.

Recent advances in deep learning have yielded remarkable success and dominate image-based tasks [8]. However, a key factor making it difficult to directly employ advanced deep architectures for video classification is that a video is naturally composed of successive images with highly overlapping content. This redundancy presents a great challenge while also offering numerous visual clues to leverage. A popular way to address this issue is to mine correlations in continuous frames. In particular, contexts at different scales in the spatiotemporal domain are crucial for recognizing videos and distinguishing similar ones.

The current leading convolutional neural networks (CNNs) for video classification [9]–[11] and their shifted version, namely 3D CNNs [12]–[14], often aggregate contexts in a later stage. More precisely, in the first layer of a typical CNN, the receptive field (RF) starts at the kernel size, which is usually small and the outputs extract only local features. As the layer goes deeper, the RF expands, and contexts begin to be utilized. These models need to be very deep to preserve rich context topologies [15] and compete with competitive trajectory-based works [3], [16]–[18]. However, simply increasing the number of layers is not wise due to parameter burden and training difficulties. In addition, these models do not embed the context evolutions of local features in the forward flow, which is essential for context mining [19]–[21]. To this end, we attempt to explore context hierarchies as early as possible for video classification.

In this paper, we first propose a novel deep alternative neural network (DANN) to mine rich context hierarchies for video classification. The DANN stacks alternative layers (ALs) consisting of a volumetric convolutional layer and a recurrent layer. The alternative deployment is used to preserve the contexts of

Manuscript received February 28, 2018; revised May 14, 2018 and June 5, 2018; accepted July 4, 2018. Date of publication July 11, 2018; date of current version September 18, 2018. This work was supported in part by the Shenzhen Peacock Plan under Grant 20130408-183003656, in part by the Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality under Grant ZDSYS201703031405467, and in part by the National Natural Science Foundation of China under Grant U1613209. The associate editor coordinating the review of this manuscript and approving it for publication was Shibasaki Ryosuke. *(Corresponding author: Wenmin Wang.)*

The authors are with the Department of Electrical and Computer Engineering, Peking University, Beijing 100080, China (e-mail: cr7or9@163.com; wangwm@ece.pku.edu.cn; wgao@pku.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.2018.2855081

local features in each layer and embed their evolutions in the hierarchical feature learning procedure. We demonstrate the advantages over standard feed-forward architectures in terms of context mining. In addition, we develop a much deeper version of the DANN by introducing a vertical dropout with skip connection to stack more ALs. We explain its benefits in terms of context exploration, faster convergence and reduction in training time. Additionally, we develop a multi-scale manner to construct DANN at various scales to further exploit complex context hierarchies, which can provide a remarkable improvement. We share good practices for applying multi-scale DANN to video classification, including input-output configurations and training-testing methods. To verify the effectiveness of our method on general video classification, we contribute a new natural disaster video dataset and make it publicly available. The experimental results obtained on four large-scale datasets demonstrate the effectiveness of our method for both human activities and natural events.

The remaining content is organized as follows. Section II reviews related works and discusses their relations to our method. Section III describes the proposed approach including the architecture of the DANN, its deeper and multi-scale version, and its application to video classification, including detailed network configurations and training methods. The experimental results and analysis are given in Section IV. Finally, Section V concludes this paper.

II. Related Works

There is a long history of video classification [22]–[31]. Early approaches interpret a video as a set of spatiotemporal points and perform video classification using these points [32] . In the last few decades, local spatiotemporal features [33] aggregated with BoF [34] or Fisher vector representations [35] have become the mainstream with state-of-the-art performances achieved on many datasets [18], [36]. Recently, with deep learning gradually dominating image-based tasks [37], there has been a trend of shifting advanced deep neural networks from image-based tasks to the video domain. In the following, we review relevant works from the perspectives of handcrafted solutions and deep-learning solutions. Particularly, in the deep-learning section, we cover two aspects of modern advances similar to our proposed method and discuss their relations.

A. Handcrafted Solutions

The standard approach to video classification with handcrafted solutions involves three major stages [18]: First, local visual features that describe a region of a video are extracted either densely [38] or sparsely [39]. Next, the features are combined into a fixed-sized video-level description using quantization techniques [34]. Finally, a classifier is trained on video-level representation to distinguish among the visual classes of interest. In particular, [39] first extended the 2D Harris corner detector to obtain representative tubes in 3D space. Since then, many 2D local descriptors have been extended to 3D to achieve video understanding, such as 3D SURF [40], HOG3D [41] and 3D SIFT [42]. The comprehensive evaluation presented in [36] compared different STIP detectors and descriptors. The authors concluded that the performance of STIPs is dataset-dependent. Afterwards, [43] made use of point trajectories to obtain the well-known dense trajectory (DT), using rich low-level descriptors to construct effective video representations. An improved version of DT was developed in [18] (iDT) to estimate camera motion, with state-of-the-art results obtained on a variety of benchmarks. Although local handcrafted features yield promising results, the major limitation is that they lack a semantic and discriminative capacity. To overcome this issue, several mid-level and high-level video representations have been proposed, such as Action Bank [44] and Dynamic-Poselets [45]. Many recent competitive works have shown that hand-crafted features [11], [46]–[48], mid-level [49], [50] and high-level [15], [51] video representations can contribute to video classification with deep neural networks.

B. Deep-Learning Solutions

Since the breakthrough in image classification with deep learning at ILSVRC 2012 [37], [52], many works have been devoted to designing effective deep networks for video classification [53]–[57]. An early work [9] extended the standard CNN to 3D for video classification but examined the proposed models on only small datasets, achieving lower performance than that of traditional features [18]. [10] designed two stream CNNs containing spatial and temporal nets by exploiting pre-trained models and an optical flow calculation. [13] investigated 3D CNNs [12] on realistic and large-scale video classification datasets. Meanwhile, several works [15], [58], [59] have tried to model long-term temporal information. However, unlike image-based tasks, deep learning does not yield significant improvements in terms of video classification over traditional methods such as the notable iDT [43]. We argue that two reasons account for this fact. First, most public video classification datasets such as UCF101 [5] and HMDB51 [6] have much smaller scales than ImageNet, in term of both the numbers of samples and categories. Second, videos are weakly labeled at the video level because of the prohibitive cost of producing detailed spatial-temporal annotations. The existing works fail to directly extract video-level representations due to the lack of context exploration. We address this issue by collecting contexts immediately after feature extraction in each layer. We insert recurrent layers among convolutional layers and stack them in an alternative way, preserving contexts from the beginning and their evolutions in a hierarchical manner. In the following, we review two relevant aspects of modern advances in deep learning that are similar to our method.

Combination of CNN and RNN. The successes of CNN and RNN have inspired researchers to exploit their combination. Most existing works focused on image-based tasks by setting CNN responsible for local feature extraction and using RNN to exploit the relative information in a handcrafted timeline. For example, [60] proposed using a convolutional RNN to learn the spatial dependencies between image regions to enhance the discriminative power of image representation. Another similar work considered using a recursive CNN to deal with temporal or sequential data and showed promising results [61]. In [62], recursive layers with the same input and output dimensions were used, but the recursive convolutions resulted in worse perfor-

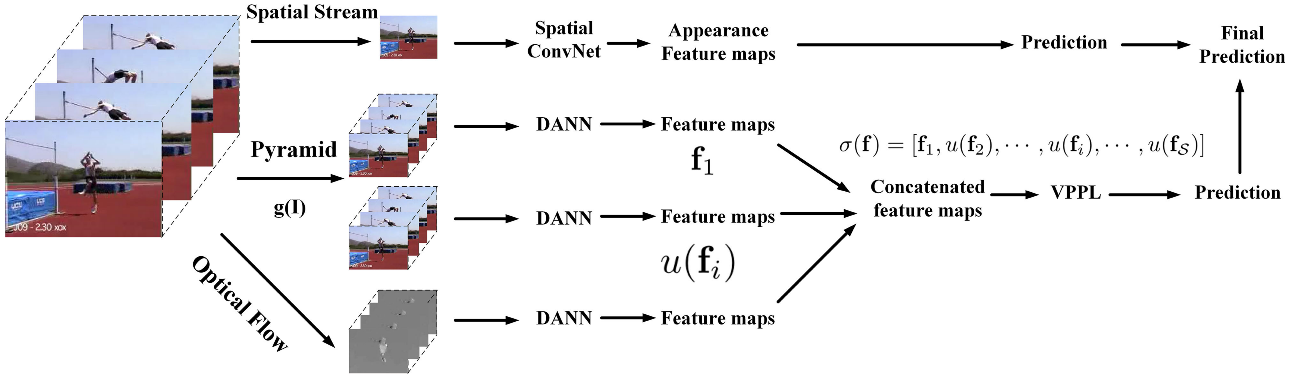

Fig. 1. Diagram of our video classification system with the multi-scale DANN. The raw input video is first cropped into a video clip of adaptive size with temporal determination and a pre-defined spatial choice. The cropped video clip is then transformed via a Laplacian pyramid. Each scale is fed to the DANN, which produces a set of feature maps. The feature maps at all scales and feature maps from the optical flow maps are then concatenated, the coarser scale maps being upsampled to match the size of the finest scale map, to generate the combined representation. The output feature maps are of arbitrary size and sent to a proposed volumetric pyramid pooling layer to be conformed to a fixed size, which is prepared for the fully connected layer to produce the predicted label. This prediction is combined with the spatial stream to produce the final prediction.

mances than that of a single convolution due to over-fitting. To overcome this over-fitting problem, [21] used a recurrent layer that takes feed-forward inputs into all unfolded layers. Our network is similar to the above works in the sense that we take advantage of CNN for local feature extraction and RNN for temporal summarization of the extracted features. However, we insert an RNN into each layer of a CNN to force the contexts of local features to be involved in the feed-forward propagation.

Going deeper in neural networks. Similar to popular deep neural networks, our DANN can be improved by going deeper. Many studies have been devoted to architecture design and training strategy of very deep networks. Earlier works adopted greedy layer-wise training or better initialization schemes to alleviate the vanishing gradients and diminishing feature reuse problems [63]. Recently, several authors introduced extra skip connections to improve information flow during forward and backward propagation. Highway networks [64] allow earlier features to flow unimpeded to later layers via parameterized skip connections, which can cross several layers at once. The skip connection parameters, learned during training, control the amount of information allowed on these highways. Similarly, residual connections were introduced in [65], in which the author gave convincing theoretical and practical evidence verifying the advantages of utilizing the additive merging of signals for object detection. They simplified highway networks by allowing shortcuts with identity functions. The authors argued that residual connections are inherently necessary to train very deep convolutional models. The deeper version of our DANN also uses identity functions to construct skip connections. In contrast, we use a vertical dropout implementation to control how to achieve skip connections.

III. Approach

A. Overview

Figure 1 decipts the diagram of our video classification system with multi-scale DANN. The raw input video is first cut adaptively in the temporal domain and pre-defined in the spatial domain (see details in Section III-E). The cropped video clip is then transformed via a Laplacian pyramid. Each scale is fed to the DANN which produces a set of feature maps. These feature maps at all scales are concatenated, the coarser scale maps being upsampled to match the size of the finest scale map. The output feature maps are of arbitrary size and sent to a proposed volumetric pyramid pooling layer (VPPL) (Section III-F) to be conformed to a fixed size, which is prepared for the fully connected layer (FCL) and softmax layer to produce the predicted semantic label.

In the following, we first introduce our deep alternative neural network, including its architecture, working principles, deeper version and multi-scale construction. Then, we provide a novel approach for adaptively determining the network input and its corresponding solutions to cope with the arbitrary size of the feature maps before FCLs. Finally, we present the training strategies.

B. Deep Alternative Neural Network

The DANN comprises a series of AL, as shown in Figure 2 (left). Between neighboring ALs, only feed-forward connections exist. Max pooling layers are optionally interleaved between the ALs. The number of recurrent iterations is set to $\mathtt{T}$ for all ALs. In the following, we present the architecture of an AL and show its advantages over the popular volumetric convolutional layer (VCL), which is used in standard 3D CNNs in terms of its ability to preserve larger RF effectively, facilitating the capture of spatiotemporal context hierarchies for video classification.

Alternative Layer. An AL consists of a VCL followed by a designed recurrent layer (RL) in an alternative manner, as shown in Figure 2 (right). Volumetric convolution is first performed to extract features from local spatiotemporal neighborhoods on the feature maps of the previous layer. Then, a recurrent layer is applied to the output and iterated $\mathtt{T}$ times. This procedure makes each unit evolve over discrete time steps and aggregate larger RFs, consuming only an extra set of layer-shared parameters of recurrent filters. Specifically, the input of a unit at position (x, y, z) in the jth feature map of the ith AL at time t, denoted

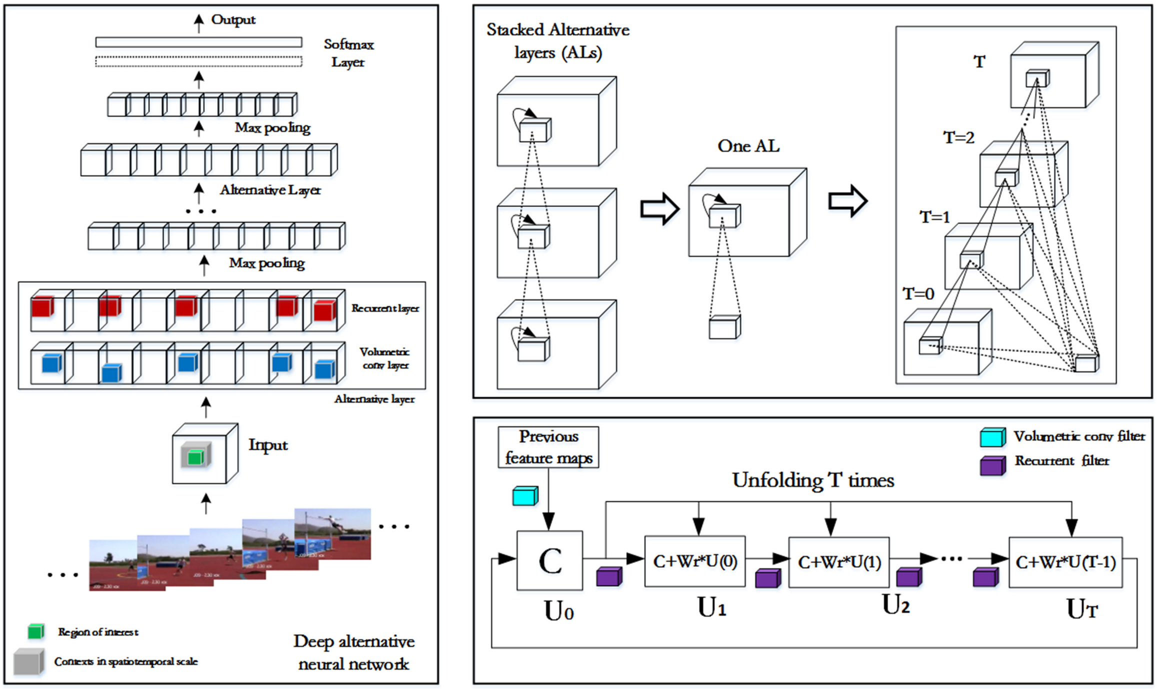

Fig. 2. The structure of the DANN (left) and its key module AL (right). The AL takes advantage of the CNN to conduct local feature extraction and the RNN to carry out temporal summarization of the extracted features. The alternative manner encourages the involvement of local feature contexts in feed-forward computations, enabling the DANN to preserve rich context hierarchies.

as $\mathbf{u}_{ij}^{xyz}(t)$, is given by

$$\mathbf{u}_{ij}^{xyz}(t) = \mathbf{u}_{ij}^{xyz}(0) + \mathbf{w}_{ij}^{r} \cdot \mathbf{u}_{ij}^{xyz}(t-1) + \mathbf{b}_{ij} \qquad (1)$$

where $\mathbf{u}_{ij}^{xyz}(t-1)$ is the recurrent input of the previous time step, $\mathbf{w}_{k}^{r}$ is the recurrent kernel, $\mathbf{b}_{ij}$ is the bias for the jth feature map in the ith layer, and $\mathbf{u}_{ij}^{xyz}(0)$ denotes the feed-forward output of the VCL

$$\mathbf{u}_{ij}^{xyz}(0) = f\left(\mathbf{w}_{(i-1)j}^{vc} \cdot \mathbf{u}_{(i-1)j}^{xyz}\right) \qquad (2)$$

$\mathbf{w}_{k}^{vc}$ is the feed-forward volumetric convolutional kernel, and f is defined as the rectified linear unit (ReLU) function followed by batch normalization (BN) [66].

The first term in Eq. 1 is the output of the volumetric convolution of the previous feature map, and the second term is induced by the recurrent connections. The state of this unit is a function of its net input is $x_{ij}^{xyz}(t) = f(\mathbf{u}_{ij}^{xyz}(t))$. Eq. 1 and Eq. 2 describe the dynamic behavior of the AL, in which the contexts are involved after the local features are extracted. Unfolding this layer over $\mathtt{T}$ time steps results in a feed-forward subnetwork of depth $\mathtt{T}+1$, as shown in Figure 2. While the recurrent input evolves over several iterations, the feed-forward input remains the same in all iterations. When $t = 0$, only the feed-forward input is present.

Advantages over VCL. The recurrent connections in AL provide three major advantages compared with VCL, which is the standard module used in C3D networks.

- The recurrent connections increase the network depth while keeping the number of adjustable parameters constant by weight sharing. Note that simply increasing the depth of VCL by sharing the weights between layers can result in the same depth and the same number parameters as DANN, as Figure 3 shows. However, this model may not be competitive with the DANN as verified in our experiments (see Table III(c) in Section IV-D).
- The structure of the AL enable every unit in the feature map to be modulated by other units in the same layer, which enhances the capability to capture statistical regularities. As the time step increases, the state of every unit is influenced by other units in an increasingly larger neighborhood in the current layer, as expressed by Eq. 1. This enables every unit to incorporate contextual information in an arbitrarily large region in the current layer. As a consequence, the

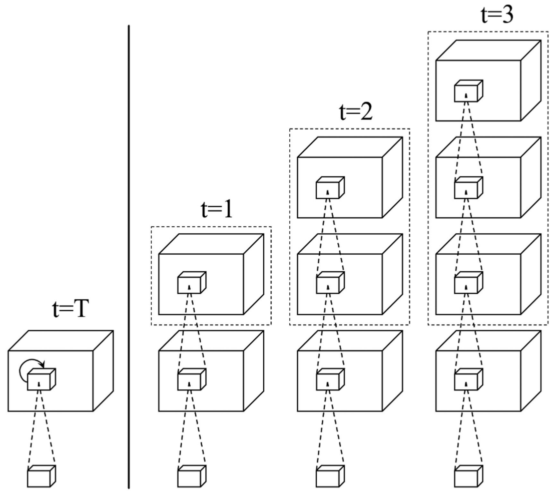

Fig. 3. Comparison of AL (left) and VCL with the same depth (right).

size of the regions that each unit can watch in the input space increases. In a VCL, the size of the RFs of the units in the current layer is fixed, and watching a larger region is possible only for units in higher layers. Unfortunately the context seen by higher-level units cannot influence the states of the units in the current layer without top-down connections [67].

- The time-unfolding mechanism is that through which each AL is interpreted as a VCL with multiple paths between the input and output. This mechanism is useful for learning complex co-adaption of local contexts and their evolutions. The recurrent connections facilitate incorporating the contextual information of each layer to refine the feature maps and, as a result, remove noisy feature activations and cause the final feature maps to be more focused on a few meaningful regions. On the other hand, the existence of shorter paths may facilitate gradient backpropagation during training [21]. Multi-path is also used in [68]–[70], but extra objective functions are used in hidden layers to alleviate the difficulty of training deep networks, which are not used in the DANN.

C. Going Deeper With Skip Connections

Many recent works have suggested that going deeper with convolutions often yields performance improvement [69]. However, experiments suggest that simply adding ALs does not result in performance improvement. This is perhaps due to that current optimization techniques do not have sufficient power to optimize a large number of layers. In this subsection, we provide a strategy for going deeper using skip connections to release the full potential of the DANN. The proposed strategy is designed in a similar fashion as that of dropout [71] but is performed vertically rather than horizontally. We expect to preserve a large network during testing but a small network during training, enjoying the advantages of dropout and knowledge distillation [72].

We use skip connections to achieve the above goal, which is motivated by a recent work [73]. We stack ALs to obtain a much deeper DANN, shrinking the depth of the DANN during training, and keeping it unchanged during testing. We randomly drop ALs during training and bypass their transformations via skip connections and identity transformation. Specifically, let p_l denote the probability that the lth AL will be used. With this definition, we can update the rule for each AL as

$$\mathbf{U}_{l+1} = \phi(p_l)\mathcal{F}(\mathbf{U}_l) \tag{3}$$

where $\mathbf{U}_l$ is the feature map of the lth AL, the units u of which are computed using Eq. 1, $\mathcal{F}$ denotes the series of computations performed in each AL, including volumetric convolution, recurrent computation, ReLU and BN, and $\phi(\cdot)$ indicates a binary choice with p_l probability that the output of the $l+1$th AL uses the information of the lth AL for each training step. In fact, when p_l equals to 1, only one AL is stacked. The comparison of the standard horizontal dropout and our vertical dropout is described in Figure 4.

To ensure the non-negative property of the AL input, we equip skip connections with an identity transformation to keep the value of $\mathbf{U}$ reasonable when $\phi(p_l)$ equals to 0. Meanwhile, we utilize skip connections to alleviate the impact of “diminishing feature reuse”, which is widely considered in very deep neural networks [64], [74], [75]. When combined with identify transformation, the update rule for each AL becomes

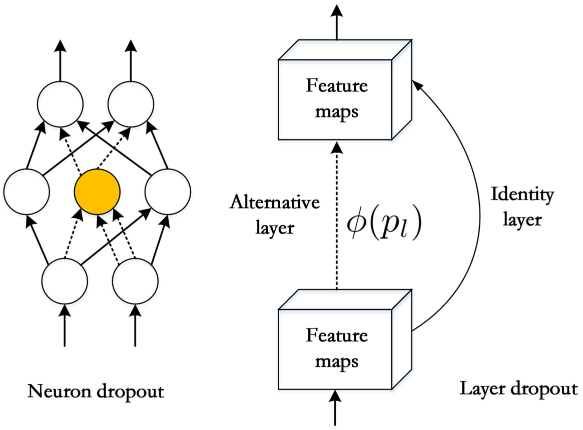

Fig. 4. Illustration of the updated version of the DANN, which goes deeper through the use of skip connection (right). To avoid over-fitting we train many sub-DANNs and use the entire DANN for testing by randomly dropping layers with an identity function, which also prevents the impact of “diminishing feature reuse”. Compared with the widely used dropout method (left), our method can be regarded as layer-wise dropout, which is performed vertically.

$$\mathbf{U}_{l+1} = \phi(p_l)\mathcal{F}(\mathbf{U}_l) + \mathbf{w}_s\mathbf{U}_l \tag{4}$$

where $\mathbf{w}_s$ is a linear projection performed to match the dimensions. When p_l equals to 0, only one identity layer is stacked.

When training a deeper DANN with skip connections, the neurons in the AL that are skipped do not contribute to the forward pass and do not participate in backpropagation. Thus, every time a mini-batch is presented, the DANN samples a different architecture of a sub-DANN. However, all these architectures share weights. This setting forces the DANN to learn more complex hierarchical contexts, which is expected to be useful for distinguishing similar actions and events. At test time, we use all ALs but multiply their outputs by p_l

$$\hat{\mathbf{U}}_{l+1} = p_l\mathcal{F}(\hat{\mathbf{U}}_l) + \mathbf{w}_s\hat{\mathbf{U}}_l \tag{5}$$

which is a reasonable approximation to the arithmetic mean of predictive distributions produced by subnetworks [37].

The proposed skip connection in the vertical direction enjoys two advantages. First, compared with simply stacking more ALs (Figure 3 right), it reduces the chain of forward propagation steps and gradient computations in training [73], which strengthens the gradients, especially in earlier layers, during backward propagation and reduces the number of parameters. Second, DANN trained with skip connections can be interpreted as an implicit ensemble of networks of different depths which is expected to preserve rich context hierarchies with complex co-adaptations of neurons, as addressed in a series of recent knowledge distillation works [72], [76]. In practice, we set the number of layers of deeper version to 18.

We refer to the standard architecture as DANN-6 and the deeper version with or without skip connections as DANN-18 (skip) and DANN-18, respectively.

D. Multi-Scale DANN

In many videos, actions and events appear in various sizes. To capture this variability, our model should be scale-invariant. We extend our DANN to the multi-scale scenario, as shown in Figure 1. Our multi-scale DANN captures the invariant property by extending the concept of volumetric weight replication to the scale space. In particular, given a video input $\mathcal{V}$, a multi-scale pyramid of video $\mathcal{V}_s$ is constructed where $\mathcal{V}_1$ has the same size as that of $\mathcal{V}$. The multi-scale pyramid can be a Laplacian pyramid and is typically processed so that local neighborhoods have zero mean and unit standard deviation. In this way, the multi-scale DANN can be obtained with parameter θ_s by instantiating one DANN per scale s and sharing all parameters across scales $\theta_s = \hat{\theta}$, where $s \in \{1, \ldots, \mathcal{S}\}$. The feature maps in the pyramid are computed using a scaling/subsampling function g_s as $\mathcal{V}_s = g_s(\mathcal{V})$ for all scales.

For each scale s, the DANN can be described as a sequence of linear transforms interspersed with nonlinear symmetric squashing units and pooling/subsampling operators. Finally, for each video clip as a training instance, the outputs of the multi-scale DANN are combined in the last AL before FCL to produce a uniform feature map

$$\sigma(\mathbf{f}) = [\mathbf{f}_1, u(\mathbf{f}_2), \ldots, u(\mathbf{f}_i), \ldots, u(\mathbf{f}_{\mathcal{S}})] \tag{6}$$

where u is an upsampling function and $\mathbf{f}_i$ is the upsampled feature map of size $\mathcal{S}$ times the size of $\mathbf{f}_1$.

As mentioned above, weights are shared within the multi-scale DANN. Intuitively, imposing complete weight sharing across scales is a natural way of forcing the network to learn scale-invariant features and, at the same time, reduces the chances of over-fitting. The more scales used to jointly train the models, the better the representation becomes for all scales.

E. Adaptive Input Configurations

The input size of a deep neural network for video classification in the temporal domain is often determined empirically since the evaluation all the choices is difficult in practice. Previous methods often consider short video intervals up to 16 frames [9], [12], [13]. A recent work [14] argues that instances in a video usually span tens or hundreds of frames and contain characteristic patterns with long-term temporal structures. The authors use 60 frames as the temporal size of the network input and demonstrate a moderate advantage over 16 frames. Similar attempts are maded in [58], [59] where longer continuous video streams are applied to their deep networks. However, the input size is still selected in an ad hoc manner, and it is difficult to favor all video classes. We introduce an adaptive method to automatically select the most discriminative video fragments using the density of the optical flow energy. We attempt to preserve motion information and appropriate range dependencies while not breaking their semantic structures in the temporal domain.

Much evidence reveals that the motion energy intensity induced by human action exhibits regular periodicity [77]. This signal can be estimated by an optical flow computation, as shown in Figure 5, and is particularly suitable for addressing our temporal estimation for the following two reasons. First, the local minima and maxima landmarks probably correspond to characteristic gesture and motion. Second, this signal is relatively robust with respect to changes in camera viewpoint, which are very common in realistic videos. In practice, we first compute the optical flow field $(\mathbf{v}_x, \mathbf{v}_y)$ for each frame I from a video Q and define its flow energy as

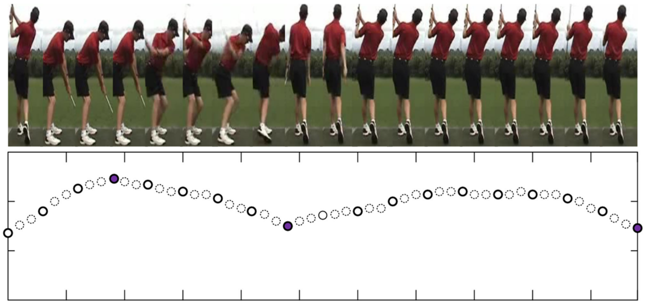

Fig. 5. Sample frames and the optical flow energy where local minima and maxima landmarks approximately correspond to motion change.

$$e(I) = \sum_{(x,y)\in\mathcal{P}} \left\| \mathbf{v}_x(x, y), \mathbf{v}_y(x, y) \right\|_2 \tag{7}$$

where $\mathcal{P}$ is the pixel-level set of selected interest points.

The energy of Q is then obtained as $\mathcal{E} = \{e(I_1), \ldots, e(I_t)\}$, which is further smoothed by a Gaussian filter to suppress noise. Subsequently, we locate the local minima and maxima landmarks $\{t\}$ of $\mathcal{E}$ and create a video fragment $\mathcal{S}$ by extracting the frames $\mathcal{S} = \{I_{t-1}, \ldots, I_t\}$ for each pair of consecutive landmarks. To address the different video clip lengths, we present a corresponding solution in Section III-F.

F. Volumetric Pyramid Pooling Layer

Typical recognition neural networks such as AlexNet [37], its deeper successors [69], [74] and shifted 3D versions [12], [13], [15], ostensibly take fixed-sized inputs. The FCLs of these networks have fixed dimensions and throw away spatial or spatiotemporal coordinates. However, our adaptive temporal decision method produces input video clips with arbitrary size. One way to generate video tube features is to feed each tube into the DANN separately. However, this tends to be time-consuming since the computations needed for overlapping tubes are not shared.

Spatial pyramid pooling improves upon the bag-of-word model in that it can maintain spatial information by pooling in local spatial bins [78], [79]. These spatial bins have sizes proportional to the image size; thus, the number of bins is fixed regardless of the image size. To adopt the DANN for input video clips of arbitrary sizes, we replace the last pooling layer with a novel VPPL inspired by the success of the spatial pyramid pooling layer [80]. Figure 6 illustrates the structure of the VPPL. In each volumetric bin, we pool the responses of each kernel (throughout this paper, we use max pooling). The outputs of the volumetric pyramid pooling are kM-dimensional vectors, where M is the number of bins and k is the number of kernels in the last AL. The fixed-dimensional vectors are then sent to the

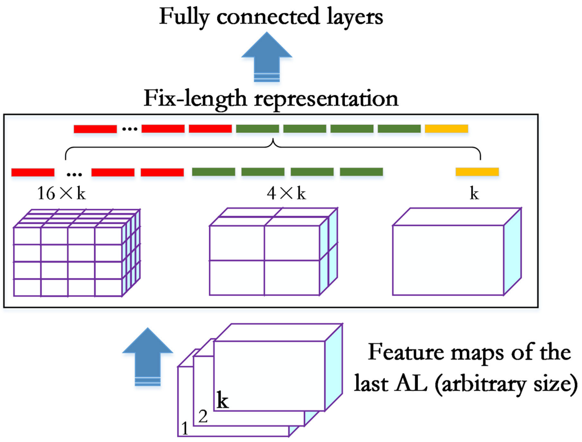
Fully connected layers
Fix-length representation
16×k
4×k
k
Feature maps of the last AL (arbitrary size)
1
2
k

Fig. 6. Structure of the volumetric pyramid pooling layer (VPPL). The function of VPPL is to resize input feature maps of arbitrary size to a fixed size.

TABLE I
SUMMARY OF CHARACTERISTICS OF NDVD

	Disaster Concept	Number
Geological Disaster	Avalanches	433
	Earthquakes	520
	Volcanic Eruptions	824
	Landslides	782
	Debris Flow	421
Hydrological Disaster	Floods	585
	Limnic Eruptions	321
	Tsunami	520
Meteorological Disaster	Blizzards	620
	Cyclonic Storms	550
	Thunderstorms	479
	Hailstorms	613
	Tornadoes	928
	Heat Waves	557
Total		8153

(a) Disaster concepts.

Disaster Concept	14
Video Number	8, 153
Mean Clip Length	15.11 seconds
Total Duration	33.9 hours
Min Clip Length	4.06 seconds
Max Clip Length	95.14 sec
Frame Rate	25 fps
Resolution	320 × 240

(b) Dataset statistics.

FCLs. With volumetric pyramid pooling, the input video clips can be of any size. This allows not only arbitrary aspect ratios but also arbitrary scales.

G. Training

Since DANN-6 and DANN-18 are special cases of the multi-scale DANN, we describe here the training procedure of the multi-scale version. The training procedure for the other two structures is the same, although without two feed-forward steps, i.e., pyramid generation and feature map combination.

To train a modern neural network for video classification, it is difficult to take the entire video as the input. Instead, we choose to train many clips for each video and then use their combination to generate the prediction for the entire video, which is widely used in recent works [11], [15]. In the clip-wise training procedure, for each video clip, the combined feature maps at all scales are sent to a softmax layer to obtain the probability of falling into the cth semantic category

$$\mathbf{y}_c = \frac{\exp(\mathbf{w}_c^\top \sigma(\mathbf{f}))}{\sum_{c'} \exp(\mathbf{w}_{c'}^\top \sigma(\mathbf{f}))}, c = 1, 2, \ldots, C \tag{8}$$

where $\sigma(\mathbf{f})$ is the concatenated output of the multi-scale DANN (Eq. 6) before the softmax layer at different scales s and $\mathbf{w}_c$ is the weight for the cth class. Finally, the loss function used in our case is the cross-entropy between the predicted probability $\mathbf{y}$ and the true hard label $\hat{\mathbf{y}}$

$$\mathcal{L} = -\sum_c \hat{\mathbf{y}}_c \log \mathbf{y}_c \tag{9}$$

where $\hat{\mathbf{y}}_c$ is set to 1 if this input clip is contained in a video annotated with the cth category and to 0 otherwise. In practice, we normalize the label vector $\hat{\mathbf{y}}$ with 1-norm $\bar{\mathbf{y}} = \hat{\mathbf{y}} / \|\hat{\mathbf{y}}\|_1$ and then use this normalized label vector to calculate the cross-entropy loss. Training is performed by minimizing the cross entropy loss function using the standard backpropagation through time (BPTT) algorithm [81], which is equivalent to using the BP algorithm for the unfolded ALs. The final gradient of a shared weight is the sum of its gradients over all time steps $\mathtt{T}$.

IV. EXPERIMENTS

A. Datasets and Setup

Natural disaster video dataset. The natural disaster datasets are rare and difficult to collect; the only one we can found is [82], which has several limitations: 1) It is composed of only 80 videos and 5 disaster concepts; 2) It views disaster detection as image classification instead of video classification, thus considering only key frames; 3) It is not publicly available.

To promote the researches on extreme event video analysis, we develop a new large-scale natural disaster video dataset (NDVD) and make it publicly available. This NDVD contains 8.15 K video and 33.9 hours of video data from YouTube and Youku, with each video lasting 15 seconds on average. The videos have a spatial resolution of 320 × 240 pixels and a 25 fps frame rate. The NDVD includes a total of 3.05 M frames distributed among 14 categories, which can be further divided into three types: Geological Disasters, Hydrological Disasters, and Meteorological Disasters. To further increase the amount of data, we perform data augmentation by randomly cropping each video shot into 8 transformed videos, each sharing the same semantic label. The summary of NDVD characteristics is shown in Tables I(a) and I(b). Figure 7 depicts a key frame sample extracted from the videos for each disaster concept.

The **UCF101** dataset [5] is a widely used benchmark with 13 K, 27 hours of video data from YouTube videos, lasting 7 seconds on average. The total number of frames is 2.4 M distributed among 101 categories. The entire dataset was split into training and testing samples three times, each split randomly selecting two-thirds of the data for training and the remaining data for testing.

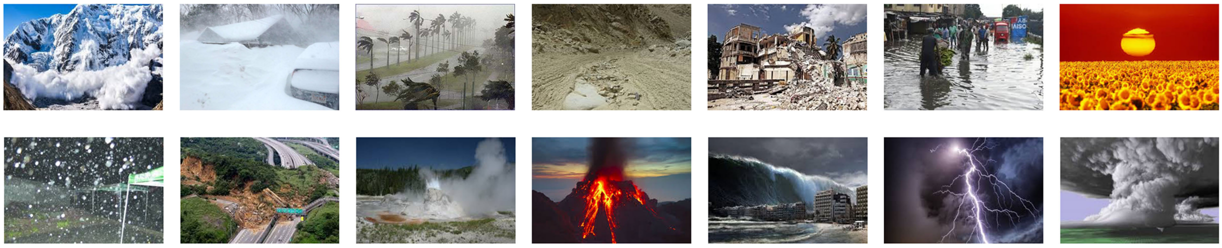

Fig. 7. NDVD sample frames. Each frame denotes a different disaster concept. From left to right (top): Avalanches, Blizzards, Cyclonic Storms, Debris Flow, Earthquakes, Floods, and Heat Waves. From left to right (down): Hailstorms, Landslides, Limbic Eruptions, Volcanic Eruptions, Tsunamis, Thunderstorms, and Tornadoes.

Fig. 8. Sample frames of three datasets used in our experiments. From top to bottom: UCF101, HMDB51, and ActivityNet.

The **HMDB51** dataset [6] contains simple facial actions, general body movements and human interactions. It is composed of 6,849 videos divided into 51 categories. We follow the original protocol using three training-testing splits. For every class and split, there are 70 videos for training and 30 videos for testing. We report the average accuracy over the three splits as the performance measure.

The **ActivityNet** dataset [7] comprises 28 K videos in 203 activity categories collected from YouTube. It consists of 68.8 hours of temporal annotations among 849 hours of untrimmed, unconstrained video. There are 1.41 action instances per video and 193 instances per class. We perform the task of *trimmed classification* using the official training set of ActivityNet v1.2, following the setup provided in the original paper [7]. Sample frames of these three datasets are shown in Figure 8.

Evaluation metrics. We use two evaluation metrics widely used in the literature, e.g., [14]. The first one is **clip-level accuracy(%)**, where we assign to each clip the class label with the maximum softmax output and measure the number of correctly assigned labels over all clips. The second metric is **video-level accuracy(%)**, which is also the standard evaluation protocol. To obtain a video score, we average the clip softmax scores and take the maximum value of this average as the class label. We average the values for all videos to obtain the video accuracy. The first metric is used to pursue a good practice of our method, while the second, used to ensure a fair comparison, is prepared for our best model to compare with other results. In our ablation experiments, we use the UCF101 split 1 dataset.

B. *Implementation Details*

The structure of the standard DANN includes 6 ALs with 64, 128, 256, 256, 512 and 512 kernel response maps, which are first followed by a VPPL and then 3 FCLs of size 2048 each. Following [13], we use $3 \times 3 \times 3$ kernel for VCL and RLs in all 6 ALs. After each AL, the DANN includes a ReLU and a volumetric max pooling layer. The max pooling kernels are of size $2 \times 2 \times 2$. All these VCLs and RLs are applied with appropriate padding and stride. The FCLs are followed by ReLU layers and a softmax layer at the end of the DANN, which output the class scores. DANN-18 enjoys the same settings as DANN-6 but has 18 ALs. The multi-scale version has a similar structure, except it also includes a pyramid-based generation procedure and a feature map concatenated operation before the FCL.

The major implementations of the DANN, including volumetric convolutions, recurrent layers and optimizations are carried out using the Torch toolbox platform [83]. For multi-GPU training, we use a parallel training strategy installed in a single system, exploiting the data parallelism and splitting each SGD batch across several GPUs [10].

For data augmentation, inspired by random spatial cropping during training [74], we apply a similar augmentation method to the spatiotemporal dimension, which we call random clipping. During training, given an input video, we first determine its temporal size t as in Section III-E. Then, we randomly select point (x, y, z) to sample a video clip of fixed size $80 \times 80 \times t$. A common alternative is to pre-process the data using a sliding window approach to obtain pre-segmented clips of fixed size. However, this approach limits the amount of data available when the windows are not overlapped, as in [13]. Another data augmentation method that we evaluate is the multi-scale cropping method [15].

We apply SGD to mini-batches with a negative log likelihood criterion. The size of each mini-batch is set to 30. Training is performed by minimizing the cross-entropy loss function using the BPTT algorithm [81]. This is equivalent to using the standard BP algorithm for a time-unfolded network. The final gradient of a shared weight is the sum of its gradients over all time steps. On NDVD, UCF101 and HMDB51, we train our network using batch size 64 for 100, 200 and 200 epochs, respectively. On ActivityNet, train using batch size 128 for 200 epochs. The initial learning rate for networks learned from scratch is 3×10^{-3}, while it is 3×10^{-4} for networks fine-tuned from pre-trained models. The above schedule is used with a 0.9 dropout

TABLE II
COMMON SETTING EXPLORATION ON UCF101 SPLIT 1 DATASET

Input	Clip	Video
RGB	62.4	64.9
MPEG [84]	71.3	73.5
Brox [85]	76.7	77.2
TVL1 [86]	78.1	**79.6**

(a) Impact of optical flow quality.

Method	Clip	Video
Sliding win.	75.4	74.8
Random clip.	78.5	79.6
Multi-scale clip.	81.2	**82.4**
Combined	81.6	82.3

(b) Impact of data augmentation.

Length	Clip	Video
16-frame	77.2	77.6
32-frame	77.3	77.2
64-frame	79.7	80.1
Adaptive	82.8	**83.0**

(c) Impact of temporal length.

Method	Clip	Video
UCF101 (S)	80.2	81.5
HMDB51 (S)	56.4	58.6
HMDB51 (F)	83.7	83.8
UCF101 (F)	62.5	65.1

(d) Impact of additional training data.

ratio. The momentum is set to 0.9, and the weight decay is initialized to 5×10^{-3} and reduced by a factor of 10^{-1} every time the learning rate decreases.

At test time, the temporal estimation t is applied to a video, and the video is divided into $80 \times 80 \times t$ clips with a temporal stride of 4 frames, where t is the adaptive temporal size. Each clip is further tested with 10 crops, namely, 4 corners and the center, together with their horizontal flips. The video-level score is obtained by averaging all the clip-level scores and crop scores. We use the clip-level accuracy to ensure that our method can be regarded as a good practice and report the video-level accuracy for comparison.

C. Common Setting Exploration

We first conduct several ablation experiments to explore the best common settings under which our model can be applied for video classification using DANN-6.

Optical flow quality. The impact of the input flow quality is summarized in Table II(a). We observe that sparse optical flow consistently outperforms RGB. The use of TVL1, as suggested in [15] yields an almost 20% increase in performance. This demonstrates that video classification is easier to learn from motion information than from raw pixel values. Given these results, we choose the TVL1 optical flow for all remaining experiments in this paper.

Data augmentation. Table II(b) demonstrates the influence of data augmentation. Our baseline is a sliding window with a 75% overlap. On the UCF101 split 1 dataset, we find that random clipping and multi-scale clipping both outperform the baseline and their combination can further boost the performance. Thus, we use the combination strategy in the following experiments.

Cross-modality pre-training. Pre-training turns out to be an effective way to initialize deep ConvNets when the target dataset does not have a sufficient number of training samples [10]. As spatial networks take RGB images as input, it is natural to exploit models trained on the ImageNet as initializations. Other modalities such as the optical flow field and RGB difference essentially capture different visual aspects of video data, and their distributions are different from those of RGB images. We used the same cross-modality pre-training technique as in [15].

Gains from adaptive temporal length. Another issue we discuss is that our DANN takes video clips with adaptive temporal length, which is different from most existing architectures for video classification. We examine this setting by comparing 6AL_VPPL_3FC with a new architecture called 6AL_3FC using fixed-size temporal lengths of 16-frame, 32-frame and 64-frame while removing the VPPL. The performance gain achieved by 6AL_VPPL_3FC on the UCF101 split 1 dataset is approximately 4.2%, as shown in Table II(c). This result verifies the advantages of our adaptive method for network temporal input.

Combining with spatial stream. A recent work [14] has demonstrated that combining appearance information learned from spatial stream can improve the performance of the 3D CNN. We examine this issue and train a network with static RGB frames in a manner similar to that in [10] by inputting 256×256 frames and cropping them randomly into 224×224 regions. The VGG-16 network [74], pre-trained on ImageNet, is fine-tuned on UCF101 and HMDB51 separately. Following the good practice in [15], we apply a weighted averaging of 0.4 and 0.6 for RGB and DANN scores, respectively.

Additional training data. We conduct experiments to see if our spatio-temporal features learned on one dataset can help to improve the accuracy for another one. The use of additional data is already known to improve results [10]. The baseline performance is 56.4%, while fine-tuning HMDB51 using UCF101 boosts the performance to 62.5%. A similar conclusion can be drawn from Table II(d). We conclude that one can learn generic representations using the DANN.

D. DANN Investigation

We investigate the optimal configurations of the DANN and demonstrate the advantages of different components. The experiments are conducted under the previous common settings.

Different configurations of DANN-6. Two crucial configurations exist for the standard DANN-6 model. The first is the AL settings, including its order and number. The other is the unfolding time T in recurrent layers. Table II shows the details of the performance comparison, where VCL is the standard volumetric convolutional layer and B_6VCL_3FC is a baseline composed of similar configurations with DANN but without ALs and an adaptive input size choice. The first column of Table III(a) has only one AL layer, and the accuracy comparison demonstrates the benefits of *exploring contexts as early as possible*. The right column of Table III(a) shows the performance gains as the number of ALs increases, revealing the advantages of the inserted recurrent layer. Table III(b) uses 6AL_VPP_3FC to study the impact of T; the results prove that larger T leads to better performance. This outcome is perhaps due to the larger contexts embedded in the DANN being more suitable for capturing semantic information. We also examine the effect of the recurrent layer implementations, by using LSTM implementations with different layers. The results in the last two lines of Table III(a) indicates that our recurrent layer implementations are better choices than complex LSTM implementations.

AL analysis. We also analyze the properties of inserting recurrent connections in an AL by comparing it with two models.

- The first one is constructed by removing the recurrent connections in each AL but adding a cascade of duplicated VCLs, denoted as **Deeper-VCL-6**, as shown in Figure 3

TABLE III
PERFORMANCE COMPARISON FOR DIFFERENT CONFIGURATIONS OF THE DANN ON THE UCF101 SPLIT 1 DATASET

Architecture	Clip	Video	Architecture	Clip	Video
B_6VCL_3FC	80.2	80.6	2AL_4VCL_VPP_3FC	85.9	85.1
AL_5VCL_VPP_3FC	85.1	86.2	3AL_3VCL_VPP_3FC	86.7	87.2
VC_AL_4VCL_VPP_3FC	83.3	84.1	4AL_2VCL_VPP_3FC	86.4	86.5
2VC_AL_3VCL_VPP_3FC	82.4	82.0	5AL_VCL_VPP_3FC	87.5	88.0
3VC_AL_2VCL_VPP_3FC	82.7	83.5	6AL_VPP_3FC	88.7	**89.0**
4VC_AL_VCL_VPP_3FC	81.4	81.9	6AL_VPP_3FC,3-layer LSTM	83.1	82.9
5VC_AL_VPP_3FC	80.9	81.5	6AL_VPP_3FC,5-layer LSTM	82.3	82.5

(a) Impact of the order and the number of AL using T = 3 in DANN-6.

Architecture	Clip	Video
6AL_VPP_3FC, T = 2	87.5	88.5
6AL_VPP_3FC, T = 3	87.2	87.9
6AL_VPP_3FC, T = 4	87.5	88.5
6AL_VPP_3FC, T = 5	88.2	88.3
6AL_VPP_3FC, T = 6	**88.7**	**89.0**
6AL_VPP_3FC, T = 2	87.6	87.8
6AL_VPP_3FC, T = 3	87.3	87.3

(b) Impact of T in DANN-6 with all ALs.

Architecture	#Param.	Clip	Video
Wider-VCL-6	3.5M	79.2	78.3
Deeper-VCL-6	3.5M	78.8	78.7
DANN-6 (T = 0)	3.5M	81.2	80.3
DANN-6 (T = 1)	3.5M	83.1	83.5
DANN-6 (T = 6)	3.5M	88.7	89.0
DANN-18	10.1M	87.5	88.2
DANN-18 (skip)	10.1M	90.4	91.9
Multi-scale DANN-18	10.1M	92.2	92.8
Multi-scale DANN-18 + spatial	10.1M	**95.1**	**95.2**

(c) Comparison with varying DANNs.

$\mathcal{S}$ \ p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	87.8	88.6	88.8	87.0	87.3	87.8	86.0	86.9	86.2	85.3
2	90.8	90.6	90.8	90.0	89.3	89.6	88.4	88.4	88.1	86.7
3	90.8	91.6	**92.8**	91.0	90.3	89.2	89.3	88.8	88.6	88.0
4	89.8	88.6	88.8	89.0	89.3	87.2	86.9	86.5	86.2	85.1
5	88.8	88.6	88.8	87.0	86.7	86.7	86.2	85.9	84.4	82.5

(d) Different probabilities of dropping each layer and scale numbers.

(right). The cascade of duplicated convolutional layers can be regarded as the time-unfolded version of the AL starting at $t = 1$ without feed-forward input. Note that both varying architectures have approximately the same number of parameters as that of the DANN.

- The second one is constructed by removing the recurrent connections in each AL, yielding a conventional 3D CNN. For fair comparison, we use more feature maps in each layer to make its number of parameters approximately the same as that of the DANN, denoted as **Wider-VCL-6**.

The performance comparison is shown in Table III(c). Clearly, our original DANN-6 achieves better performance than that of both the wider version and deeper version in terms of accuracy on the UCF101 split 1 dataset, which demonstrates the importance of inserting recurrent connections in an AL.

Going deeper with the DANN. The main factor using skip connection is the layer dropout probability p_l. In practice, we set p_l the same for each layer as p. We evaluate typical choices of p and the performance comparison is shown in Table III(d). We can see a clear advantage using different probability less than 1 compared to $p = 1$, which indicates no skip connections but adds an identify function like P3D ResNet [47]. Among various p in our experiments, the best performance is obtained in $p = 0.3$.

Multi-scale DANN. In our multi-scale solution, we mainly consider the pyramid construction manners. We use a Laplacian pyramid with various scales to examine the impact compared with a basic DANN without the multi-scale method. The performance is shown in Table III(d), which shows that the multi-scale DANN always outperforms the basic DANN, and the optimal choice for the multi-scale DANN-18 is obtained around 3.

E. Results and Analysis

Table IV reports the results of the best DANN model in terms of video-level accuracy and other state-of-the-art approaches over three splits on the UCF101 and HMDB51 datasets. We list them according to performance in Table IV. As shown, pure local features such as iDT are surpassed by most deep-learning solutions. We can also conclude that adding local features can yield performance improvement, e.g., [11], [46]–[48]. This indicates that each pure single deep network has the potential to be improved by combining local features. Note that all the other deep networks use a pre-defined temporal length to generate video clips as the input, such as 16 frames [13] and 60 frames [14], while our DANN determines it in an adaptive manner. Combined with spatial stream, DANN-18 outperforms all other methods, achieving an accuracy of 95.7% on the

TABLE IV
COMPARISON WITH THE STATE-OF-THE-ART ON HMDB51 AND UCF101

Method	Year	HMDB51	UCF101
Two-stream [10]	2014	59.4	88.0
Hidden Two-Stream [87]	2017	60.2	90.3
iDT+FV [18]	2013	57.2	85.9
iDT+HSV [3]	2016	61.1	88.0
MoFAP [88]	2016	61.7	88.3
LTC+spatial [14]	2016	61.5	88.6
KVMF [54]	2016	63.3	93.1
TDD+iDT [11]	2015	65.9	91.5
Two-Stream I3D [89]	2017	66.4	93.4
L^2STM [55]	2017	66.2	93.6
Adascan+iDT+C3D [46]	2016	66.9	93.2
SPN [51]	2017	68.9	94.6
ST-VLMPF [50]	2017	69.5	93.6
P3D ResNet + iDT [47]	2017	-	93.7
Cool-TSN [90]	2017	69.5	94.2
TSN [15]	2017	71.0	94.9
ShuttleNet [91]	2017	71.7	95.4
SMN+iDT [48]	2017	72.2	94.9
DOVF+MIFS [49]	2017	**75.0**	95.3
Wider-VCL-6		65.2	86.7
Deeper-VCL-6		66.8	88.3
DANN-6		69.3	89.2
DANN-18		71.9	93.6
Multi-scale DANN-18		73.9	95.5
Multi-scale DANN-18 + spatial		74.3	**95.7**

TABLE V
COMPARISON WITH THE STATE-OF-THE-ART ON ACTIVITYNET

Method	Year	mAP
iDT + FV [18]	2013	66.5
Two-Stream [10]	2014	71.9
C3D [13]	2015	74.1
Depth2Action [56]	2016	73.4
TSN [15]	2016	89.6
UntrimmedNet [57]	2017	91.3
Wider-VCL-6		79.1
Deeper-VCL-6		75.5
DANN-6		86.3
DANN-18		89.3
Multi-scale DANN-18		91.7
Multi-scale DANN-18 + spatial		**92.4**

TABLE VI
COMPARISON WITH OTHER METHODS ON NDVD

Method	Year	mAP
iDT + FV [18]	2013	28.5
Two-Stream [10]	2014	32.9
C3D [13]	2015	42.1
Two-Stream I3D [89]	2017	40.2
Hidden Two-Stream [87]	2017	40.7
P3D ResNet [47]	2017	41.1
ShuttleNet [91]	2017	42.2
Wider-VCL-6		46.2
Deeper-VCL-6		42.8
DANN-6		51.3
DANN-18		47.3
Multi-scale DANN-18		51.7
Multi-scale DANN-18 + spatial		**52.4**

UCF101 dataset. However, on HMDB51, the best performance corresponds to a deep local video feature with multi-skip feature stacking strategy [49]. We attribute this result to the fact that they employed advanced temporal segment network [15], which is powerful in terms of temporal segmentation. Also note that most approaches can obtain a performance gain when spatial stream is utilized, including our DANN.

For the ActivityNet dataset, we compare with recent successful video classification methods that have achieved state-of-the-art performances, including improved trajectories (iDT+FV) [18], two-stream CNNs [10], C3D [13], temporal segment network (TSN) [15], UntrimmedNet [57], and Depth2Action [56]. The results are summarized in Table V. We present the results of our DANN with four implementations at the bottom of the figure. Clearly, our shallow network DANN-6 is already competitive, achieving an 86.3% accuracy. Furthermore, our best model (DANN-18 combined with spatial stream) outperforms all these previous methods, achieving an accuracy of 92.4%. Our best results are better than that of methods by at least 1.5%. This clear improvement demonstrates the advantages of our method, especially the deeper and multi-scale design. The superior performance of our method demonstrates the effectiveness of using alternative layer for trimmed videos and the importance of effective hierarchical context mining. Potentially, we can still improve the performance by incorporating competitive local descriptors such as iDT.

For our proposed NDVD, we examine seven recent competitive methods and compare them with our method, including previously mentioned iDT+FV [18], two-stream CNNs [10], C3D [13], Two-Stream I3D [89], Hidden Two-Stream [87], P3D ResNet [47] and ShuttleNet [91]. The results are listed in Table VI. In general, we can draw conclusions similar to those for the previous three human action datasets. The best version of our architecture outperforms all the other methods. Specifically, compared with four very recent methods, which were proposed in 2017, the multi-scale DANN-18 with spatial stream can yield an at least 7.2% improvement, achieving an accuracy of 52.4%. We notice that the performance of the trajectory-based method is rather low, while deep-learning solutions generally achieve better results. This result suggests that handcrafted features may not be suitable for natural disaster videos. Note that the overall performance is much lower than that on the human action datasets, which indicates the difficulties in recognizing natural events. Our NDVD offers the video analysis community a new challenging natural video dataset that extends beyond human actions.

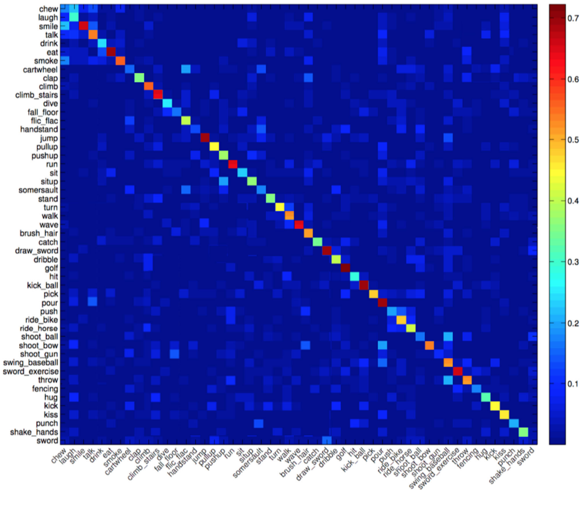

Fig. 9. Confusion matrix for DANN-6 on HMDB51 dataset.

Confusion matrix analysis. In this section, we provide a more fine-grained analysis of our method using a per-class confusion matrix on the HMDB51 dataset. We use the **DANN-6** to compare with the **Deeper-VCL-6**, which is equivalent to the standard C3D. The two architectures have approximately the parameters, but the latter removes the recurrent connections in the AL and uses a deeper VCL with the same weights (the number of additional layers is the same as the number of instances of unfolding times in each AL), as discussed in Figure 3, Section IV-D, and Table III(d). Figures 9 and 10 show that in most cases, the **DANN-6** outperforms the **Deeper-VCL-6**. For example, we can see clear advantages on “climb stairs” and “kick ball” classes. Table III(c), IV, V and VI quantitatively demonstrate the advantages of our AL and DANN structure.

F. Model Visualization

To gain insight what DANN learns, we first visualize the first-layer spatiotemporal convolutional filters in the vector-field

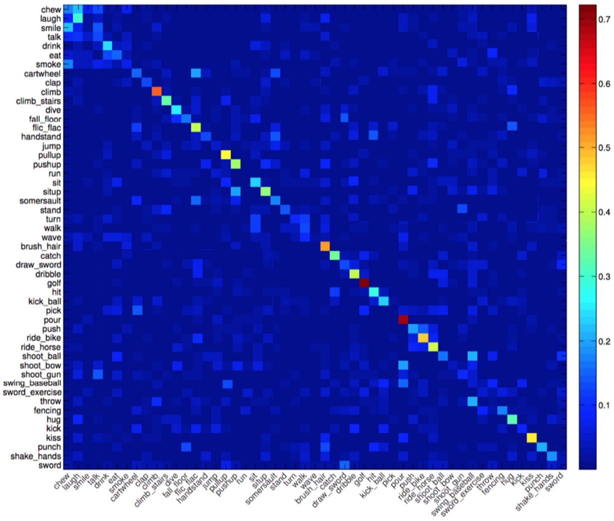

Fig. 10. Confusion matrix for Deeper-VCL-6 on HMDB51 dataset.

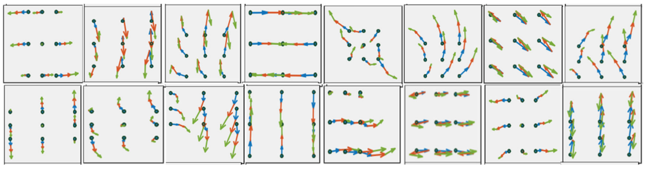

Fig. 11. Spatiotemporal filters from the first layer of the DANN learned using a 2-channel and TVL1 optical flow on UCF101. 16 out of 64 filters are presented. Each cell in the grid represents two 3 × 3 × 3 filters for 2-channel flow input (one for x and one for y). x and y intensities are converted into vectors in 2D. The third dimension (time) is indicated by putting vectors one after another in different colors for better visualization.

form. Filters learned on 2-channel optical flow vectors have dimensions of $2 \times 3 \times 3 \times 3$ corresponding to the channels, width, height and time, respectively. For each filter, we take the two channels in each $3 \times 3 \times 3$ volume and visualize them as vectors using x- and y-components. Figure 11 shows 16 example filters out of 64 from a network learned on UCF101 with TVL1 optical flow input [86]. Since our filters are spatiotemporal, they have a third dimension in time. Following the visualization method in [14], we describe these filters as vectors concatenated one after another with regard to the time step. We denote each time step using different colors and see that the filters learned by the DANN are able to represent complex motions in local neighborhoods, which enables the incorporation of even more complex patterns in later stages of the network.

For the filters in the middle layer, we use the visualization method similar to that in [14]. We illustrate example frames from top-scoring videos for a set of selected filters in the fifth layers of the DANN and C3D (the same style of Deeper-VCL that has the approximate parameters with DANN), as shown in Figure 12. Clearly, for filters maximizing the homogeneity of returned class labels, the top activations for filters of the DANN result in videos with similar action classes. The grouping of videos by classes is less prominent for activations of the C3D network. This result indicates that the DANN has higher levels of abstraction at the corresponding convolution layers when compared to the popular C3D networks. We also use another popular activation based visualization method, i.e. gradient-weighted class activation mapping (Grad-CAM) [92], which uses the gradients of any semantic class, flowing into the last convolutional layer, to produce a coarse localization map highlighting the important regions in the image for predicting the class. Figure 13 shows that the activations of our DANN can efficiently focus on the meaningful parts in the frame.

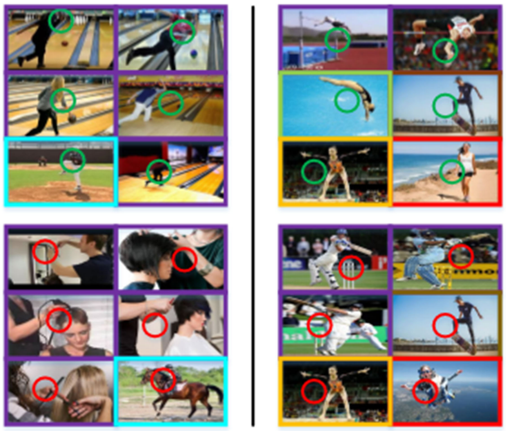

Fig. 12. Frames corresponding to videos with top activations at the fifth ALs of the DANN-6 (left) and C3D (right). Circles indicate the spatial location of the maximum response. The visualized frames correspond to the maximum response in time. DANN-6 has 4 classes out of 12 videos, while C3D has 9 classes out of 12 videos.

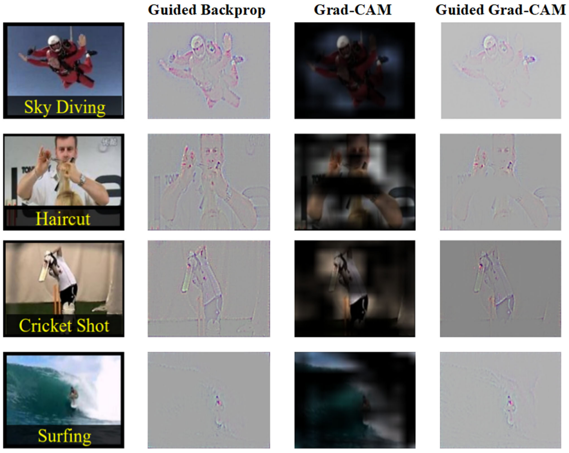

Fig. 13. The Grad-CAM visualizations of DANN-6 for frames sampled from the UCF101 test set.

In addition to filter analysis, we seek to attain further insight into the learned DANN models. In this sense, we adopt the DeepDraw toolbox as used in [15]. This tool conducts iterative gradient ascent on input images with only white noises. Thus, the output after a number of iterations can be considered as a class visualization based solely on class knowledge inside the model. The original version of the tool addresses only RGB data. To conduct visualization for optical-flow-based models, we adapt the tool to work with our DANN. As a result, we visualize interesting class information in the video classification models. We randomly select four classes from four dataset for visualization. The results are shown in Figure 14. For both

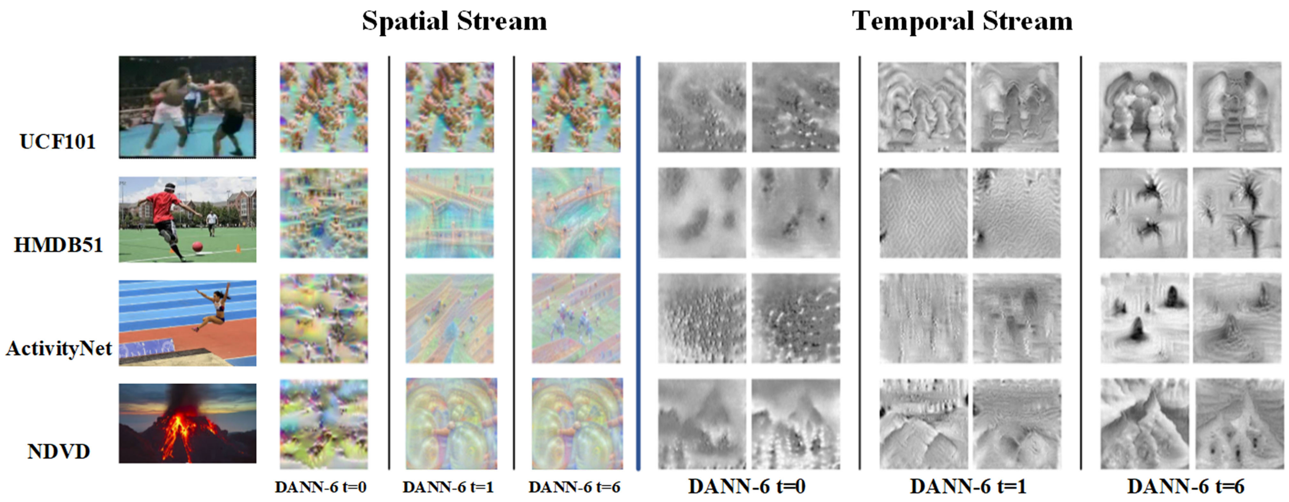

Fig. 14. Model visualization for video classification using DeepDraw [15] on four datasets. We compare three settings: (1) DANN-6 t = 0 which is equivalent to the standard C3D (2) DANN-6 t = 1; (3) DANN-6 t = 6. For spatial ConvNets, we plot three generated visualization as color images. For temporal ConvNets, we plot the flow maps of x (left) and y (right) directions in gray-scales. Note all these images are generated from purely random pixels.

RGB and optical flow, we visualize the DANN model learned under the following three settings: (1) DANN-6 t = 0 which is equivalent to the standard C3D; (2) DANN-6 t = 1; (3) DANN-6 t = 6. In this figure, we can see that DANN-6 t = 6 can capture more semantic information than the other two settings, which indicates the effectiveness of the recurrent connections in the AL.

V. Conclusion

We present a novel deep alternative neural network for video classification, which enjoys the strength of both CNN and RNN by incorporating context evolutions into the forward propagation. We show how to go deeper with skip connection and construct a multi-scale version that facilitates the collection of rich context hierarchies. To evaluate our method beyond the realm of human-centric videos, we develop a new large-scale natural disaster video dataset and make it publicly available. The experiments on four challenging benchmarks demonstrate the advantages of our model on data related to both human actions and natural extreme events.

References

[1] Y. Yang *et al.*, "IF-MCA: Importance factor-based multiple correspondence analysis for multimedia data analytics," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 1024–1032, Apr. 2018.

[2] S. Pouyanfar, Y. Yang, S. C. Chen, M. L. Shyu, and S. S. Iyengar, "Multimedia big data analytics: A survey," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–34, 2018.

[3] 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. Vis. Image Understanding*, vol. 150, pp. 109–125, 2016.

[4] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *Proc. ACM Multimedia Conf.*, 2016, pp. 791–800.

[5] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," arXiv:1212.0402, 2012.

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

[7] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 961–970.

[8] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

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

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

[11] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4305–4314.

[12] 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.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[14] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

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

[16] 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. Vis. Pattern Recognit.*, 2015, pp. 204–212.

[17] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 581–595.

[18] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3551–3558.

[19] A. Sharma, O. Tuzel, and M.-Y. Liu, "Recursive context propagation network for semantic scene labeling," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2447–2455.

[20] J. Wang, W. Wang, X. Chen, R. Wang, and W. Gao, "Deep alternative neural network: Exploring contexts as early as possible for action recognition," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 811–819.

[21] M. Liang, X. Hu, and B. Zhang, "Convolutional neural networks with intra-layer recurrent connections for scene labeling," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 937–945.

[22] S. W. Smoliar and H. Zhang, "Content based video indexing and retrieval," *IEEE Multimedia*, vol. 1, no. 2, pp. 62–72, Summer 1994.

[23] N. Dimitrova and F. Golshani, "Motion recovery for video content classification," *ACM Trans. Inf. Syst.*, vol. 13, no. 4, pp. 408–439, 1995.

[24] W. Xiong and J. C.-M. Lee, "Efficient scene change detection and camera motion annotation for video classification," *Comput. Vis. Image Understanding*, vol. 71, no. 2, pp. 166–181, 1998.

[25] G. Yu, N. A. Goussies, J. Yuan, and Z. Liu, "Fast action detection via discriminative random forest voting and top-K subvolume search," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 507–517, Jun. 2011.

[26] P. Cui, F. Wang, L.-F. Sun, J.-W. Zhang, and S.-Q. Yang, "A matrix-based approach to unsupervised human action categorization," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 102–110, Feb. 2012.

[27] S. Wang *et al.*, "Semi-supervised multiple feature analysis for action recognition," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 289–298, Feb. 2014.

[28] F. Patrona, A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Visual voice activity detection in the wild," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 967–977, Jun. 2016.

[29] 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.

[30] W. Xu, Z. Miao, X.-P. Zhang, and Y. Tian, "A hierarchical spatio-temporal model for human activity recognition," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1494–1509, Jul. 2017.

[31] B. Krüger *et al.*, "Efficient unsupervised temporal segmentation of motion data," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 797–812, Apr. 2017.

[32] S. A. Niyogi and E. H. Adelson, "Analyzing and recognizing walking figures in XYT," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 469–474.

[33] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit.*, 2004, vol. 3, pp. 32–36.

[34] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

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

[36] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 124.1–124.11.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[38] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3185–3192.

[39] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2/3, pp. 107–123, 2005.

[40] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 650–663.

[41] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. 19th Brit. Mach. Vis. Conf.*, 2008, pp. 275:1–275:10.

[42] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 357–360.

[43] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.

[44] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1234–1241.

[45] L. Wang, Y. Qiao, and X. Tang, "Video action detection with relational dynamic-poselets," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 565–580.

[46] A. Kar, N. Rai, K. Sikka, and G. Sharma, "AdaScan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5699–5708.

[47] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5534–5542.

[48] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7445–7454.

[49] Z. Lan, Y. Zhu, A. G. Hauptmann, and S. Newsam, "Deep local video feature for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2017, pp. 1219–1225.

[50] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal vector of locally max pooled features for action recognition in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3097–3106.

[51] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2097–2106.

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

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

[54] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao, "A key volume mining deep framework for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1991–1999.

[55] L. Sun *et al.*, "Lattice long short-term memory for human action recognition," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2166–2175.

[56] Y. Zhu and S. Newsam, "Depth2action: Exploring embedded depth for large-scale action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 668–684.

[57] L. Wang, Y. Xiong, D. Lin, and L. V. Gool, "UntrimmedNets for weakly supervised action recognition and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6402–6411.

[58] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 2625–2634.

[59] J. Y.-H. Ng *et al.*, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4694–4702.

[60] Z. Zuo *et al.*, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2015, pp. 18–26.

[61] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 656–664.

[62] D. Eigen, J. Rolfe, R. Fergus, and Y. LeCun, "Understanding deep architectures using a recursive convolutional network," arXiv:1312.1847, 2013.

[63] D. Erhan *et al.*, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learning Res.*, vol. 11, pp. 625–660, 2010.

[64] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2377–2385.

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

[66] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[67] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," arXiv preprint arXiv:1412.6856, 2014.

[68] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.

[69] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[70] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[71] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting." *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[72] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

[73] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 646–661.

[74] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014.

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

[76] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev training for neural networks," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4281–4290.

[77] R. Waters and J. Morris, "Electrical activity of muscles of the trunk during walking," *J. Anatomy*, vol. 111, no. 2, pp. 191–199, 1972.

[78] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1458–1465.

[79] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 2169–2178.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[81] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[82] S. Pouyanfar and S. C. Chen, "Semantic concept detection using weighted discretization multiple correspondence analysis for disaster information management," in *Proc. IEEE 17th Int. Conf. Inf. Reuse Integr.*, 2016, pp. 556–564.

[83] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A MATLAB-like environment for machine learning," in *Proc. BigLearn, NIPS Workshop*, 2011, Art. no. EPFL-CONF-192376.

[84] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2593–2600.

[85] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 25–36.

[86] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in *Proc. 29th DAGM Conf. Pattern Recognit.*, 2007, pp. 214–223.

[87] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," arXiv:1704.00389, 2017.

[88] L. Wang, Y. Qiao, and X. Tang, "MoFAP: A multi-level representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 254–271, 2016.

[89] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.

[90] C. Roberto de Souza, A. Gaidon, Y. Cabon, and A. Manuel Lopez, "Procedural generation of videos to train deep action recognition networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4757–4767.

[91] Y. Shi, Y. Tian, Y. Wang, W. Zeng, and T. Huang, "Learning long-term dependencies for action recognition with a biologically-inspired deep network," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 716–725.

[92] R. R. Selvaraju *et al.*, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

Jinzhuo Wang received the B.S. degree in 2013 from the School of Electronic Engineering and Computer Science, Peking University, Beijing, China, where he is currently working toward the Ph.D. degree. His research interests include computer vision and deep learning. He has authored or coauthored several papers on relevant conferences and journals, such as NIPS, ACM Multimedia, AAAI, ICME, ICIP, and IEEE TRANSACTIONS ON MULTIMEDIA.

Wenmin Wang (M'16) received the Ph.D. degree in computer architecture from the Harbin Institute of Technology, Shenzhen, China, in 1989. He was an Assistant Professor and an Associate Professor with the Harbin University of Science and Technology as well as the Harbin Institute of Technology. Since 1992, he has gained about 18 years of oversea industrial experiences in Japan and America, where he served as Staff Engineer, Chief Engineer, General Manager of software division, etc. Then, he joined the School of Electronic and Computer Engineering, Peking University, Beijing, China, as a Professor by the end of 2009. His research interests include computer vision, multimedia retrieval, artificial intelligence, and machine learning.

Wen Gao (M'92–SM'05–F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He is currently a Professor of Computer Science with Peking University, Beijing, China. He was a Professor of Computer Science with the Harbin Institute of Technology from 1991 to 1995, and a Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has authored or coauthored extensively including five books and more than 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He served on the Editorial Board for several journals, such as the IEEE TRANSACTIONS ON CIRUCITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, the *EURASIP Journal of Image Communications*, and the *Journal of Visual Communication and Image Representation*. He Chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE, ICME, and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations.