Multilayer and Multimodal Fusion of Deep Neural Networks for Video Classification

Xiaodong Yang Pavlo Molchanov Jan Kautz
NVIDIA
{xiaodongy, pmolchanov, jkautz}@nvidia.com

ABSTRACT

This paper presents a novel framework to combine multiple layers and modalities of deep neural networks for video classification. We first propose a multilayer strategy to simultaneously capture a variety of levels of abstraction and invariance in a network, where the convolutional and fully connected layers are effectively represented by our proposed feature aggregation methods. We further introduce a multimodal scheme that includes four highly complementary modalities to extract diverse static and dynamic cues at multiple temporal scales. In particular, for modeling the long-term temporal information, we propose a new structure, FC-RNN, to effectively transform pre-trained fully connected layers into recurrent layers. A robust boosting model is then introduced to optimize the fusion of multiple layers and modalities in a unified way. In the extensive experiments, we achieve state-of-the-art results on two public benchmark datasets: UCF101 and HMDB51.

CCS Concepts

•Information systems → Multimedia and multimodal retrieval; •Computing methodologies → Video summarization;

Keywords

Video Classification; Deep Neural Networks; Boosting; Fusion; CNN; RNN;

1. INTRODUCTION

Content-based video classification is fundamental to intelligent video analytics including automatic categorizing, searching, indexing, segmentation, and retrieval of videos. It has been applied to a wide range of real-word applications, for instance, surveillance event detection [49], semantic indexing [1], gesture control [11], and so forth. It is a challenging task to recognize unconstrained videos because 1) an appropriate video representation can be task-dependent, e.g., coarse ("swim" vs. "run") or fine-grained ("walk" vs. "run") categorizations; 2) there may be multiple streams of information that need to be taken into account, such as actions, objects, scenes, etc.; 3) there are large intra-class variations, which arise from diverse viewpoints, occlusions and backgrounds. As the core information of videos, visual cues provide the most significant information for video classification. Most traditional methods rely on the bag-of-visual-words (BOV) representation which consists of computing and aggregating visual features [13]. A variety of local and global visual features have been proposed, for instance, GIST [31] and SIFT [27] can be used to capture static information in spatial frames, while STIP [23] and improved dense trajectories (iDT) [44] are widely employed to compute both appearance and motion cues in videos.

There is a growing trend to learn robust feature representations with deep neural networks for various tasks such as image classification [20], object detection [35], natural language processing [40], and speech recognition [6]. As one of the most successful network architectures, the recent surge of convolutional neural networks (CNN) has given rise to a number of methods to employ CNN for video classification. Karparthy et al. [19] made the first attempt to use a buffer of video frames as input to networks, however, the results were inferior to those of the best hand-engineered features [44]. Tran et al. [42] proposed C3D using 3D-CNN over short video clips to learn appearance and micro-motion features simultaneously. However, these methods focus on short or mid-term information as feature representations are learned in short-time windows. This is insufficient for video classification since complex events are better described by leveraging the temporal evolution of short-term contents. In order to capture long-term temporal clues in videos, recurrent neural networks (RNN) were applied to explicitly model videos as an ordered sequence of frames [8, 29].

CNN-based video classification algorithms typically make predictions using the softmax scores or, alternatively, they use the last fully connected layer as a feature representation [36, 42], because CNN hierarchically compute abstract and invariant representations of the inputs. However, leveraging information across multiple levels in a network has proven beneficial for several tasks such as natural scene recognition [48], object segmentation [26] and optical flow computation [10]. This is somewhat expected since convolutional layers retain spatial information as opposed to fully connected layers. For video classification, we argue that appropriate levels of abstraction and invariance in CNN for video representation are also task- and class-dependent. For example,

MM '16, October 15–19, 2016, Amsterdam, The Netherlands.

ISBN 978-1-4503-3603-1/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2964284.2964297

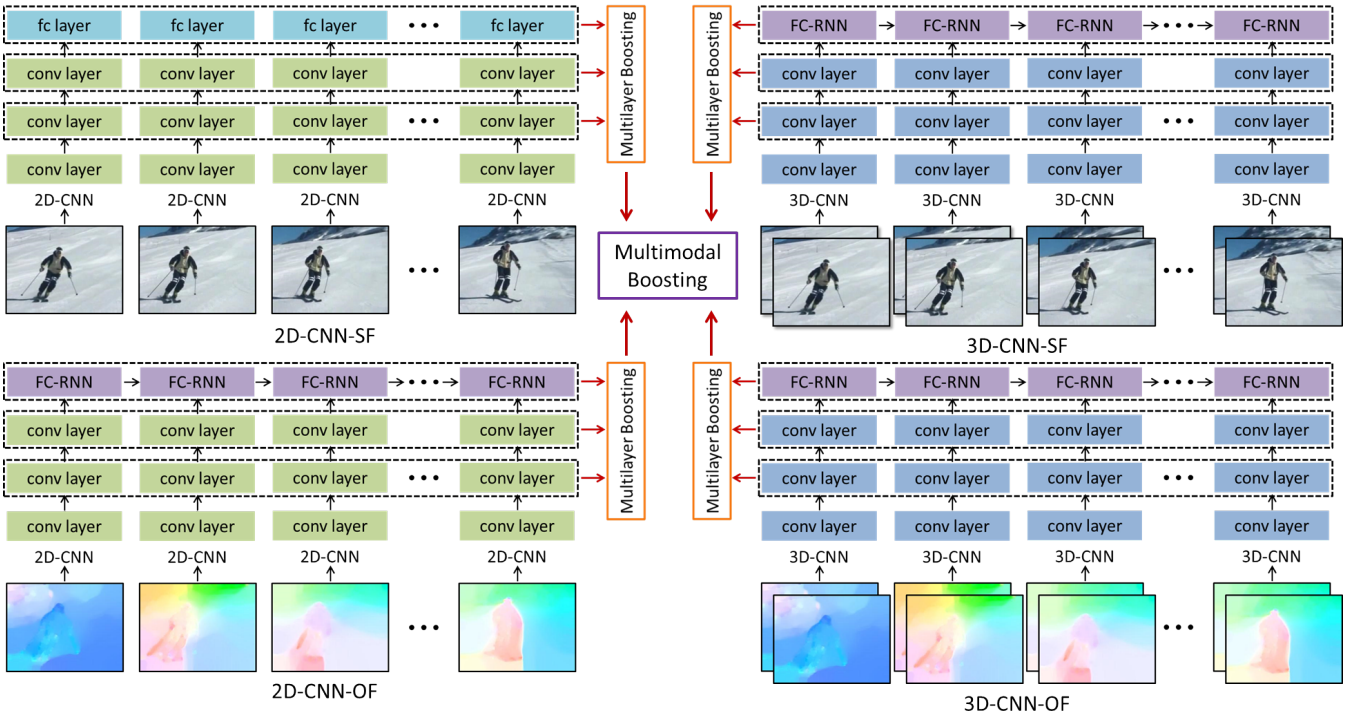

Figure 1: An overview of the proposed multilayer and multimodal fusion framework for video classification. We use four modalities to extract highly complementary information across multiple temporal scales. For each single modality, discriminative representations are computed for convolutional and fully connected layers. We employ an effective boosting model to fuse the multiple layers and modalities. Box colors are encoded according to different networks: 2D-CNN and 3D-CNN with and without RNN. We propose FC-RNN to model long-term temporal information rather than using the standard RNN structure.

distinguishing "soccer game" and "basketball game" requires high-level representations to model global scene statistics. However, classification of "playing guitar" and "playing violin" demands fine-scale features to capture subtle appearance and motion features. Therefore, leveraging the multilayer abstractions is expected to simplify video classification.

Although a significant progress in recent years has been achieved in the development of feature learning by deep neural networks [15, 20, 37], it is clear that none of the features have the same discriminative capability over all classes. For example, videos of "wedding ceremony" are strongly associated with static scenes and objects, while "kissing" is more related to dynamic motions. It is therefore widely accepted to adaptively combine a set of complementary features rather than using a single feature for all classes. Simonyan et al. [36] proposed the two-stream networks based on 2D-CNN to explicitly incorporate motion information from optical flow to complement the static per-frame information. Simple late fusion was adopted to combine the softmax scores of two networks by either averaging or with a linear classifier. This method has been widely utilized for video analysis [8, 46] thanks to the two complementary modalities and outstanding performance. Nevertheless, the question of which robust modalities to exploit and how to effectively perform multimodal fusion still remains open for video classification.

In this paper, we propose a multilayer and multimodal fusion framework of deep neural networks for video classification. The multilayer strategy can simultaneously capture a variety of levels of abstractions, thus is able to adapt from coarse- to fine-grained categorizations. Instead of using only two modalities as in the two-stream networks [36], we propose to use four complementary modalities in our multimodal scheme, i.e., 2D-CNN on a single spatial (color) frame and optical flow image as well as 3D-CNN on a short clip of spatial (color) frames and optical flow images. They not only effectively harness cues about static objects and dynamic motions in videos but also effectively exploit across temporal scales. As for the fusion of multiple layers and modalities, we adopt a powerful boosting model to learn their optimal combination.

Fig. 1 illustrates the overview of our proposed multilayer and multimodal fusion framework. Given an input video, the four modalities are used to extract complementary information at short and mid-term temporal scales. Instead of using the standard RNN structure, we propose FC-RNN to model the long-term temporal evolution across a whole video. FC-RNN takes advantage of pre-trained networks to transform the pre-trained fully-connected (`fc`) layers into recurrent layers. In the following, we use 2D-CNN-SF, 2D-CNN-OF, 3D-CNN-SF, 3D-CNN-OF to indicate 2D-CNN and 3D-CNN on spatial (color) frames and optical flow, respectively. For each individual network, an improved Fisher vector (iFV) is proposed to represent convolutional (`conv`) layers and an explicit feature map is used to represent the `fc` layers. We then employ a robust boosting model to learn the optimal combination of multiple layers and modalities. The main contributions of this paper are summarized as follows:

- We present a multilayer fusion strategy to capture multiple levels of abstraction and invariance in a single network. We propose to use the iFV and explicit feature maps to represent features of `conv` and `fc` layers.

- We introduce a multimodal fusion scheme to incorporate four highly complementary modalities to extract static and dynamic cues from multiple temporal scales. In particular, we propose FC-RNN to preserve the generalization properties of pre-trained networks.

- We adopt an effective boosting model for video classification by fusing multiple layers and modalities in an optimal and unified way.

- In the extensive experiments, our method achieves superior results on the well-known UCF101 and HMDB51 benchmarks.

2. RELATED WORK

Videos have been studied by the multimedia community for decades. Over the years a variety of problems like multimedia event recounting, surveillance event detection, action search, and many more have been proposed. A large family of these studies is about video classification. Conventional video classification systems hinge on extraction of local features, which have been largely advanced in both detection and description. Local features can be densely sampled or selected by maximizing specific saliency functions. Laptev [23] proposed STIP to detect sparse space-time interest points by extending the 2D Harris corner detector into 3D. Wang et al. [44] introduced the improved dense trajectories (iDT) to densely sample and track interest points from multiple spatial scales, where each tracked interest point generates a set of descriptors to represent shape and motion. Many successful video classification systems use iDT together with the motion boundary histogram (MBH) descriptor, which comprises the gradient of horizontal and vertical components of optical flow. It is widely recognized as the state-of-the-art feature for video analysis.

After local feature extraction, a number of coding techniques have been proposed for feature quantization, e.g., sparse coding [28] and locality-constrained linear coding [45]. Then average pooling and max pooling are normally used to aggregate statistics from local features. Several more advanced coding methods, e.g., Fisher vector (FV) [34] and vector of locally aggregated descriptors (VLAD) [16], have emerged to reserve high order statistics of local features and achieve noticeably better performance. However, these methods obviously incur the loss of spatio-temporal order of local features. Extensions to the completely orderless aggregation methods include spatio-temporal pyramids [24] and super sparse coding vectors [50]. Graphical models, such as hidden Markov models (HMM) and conditional random fields (CRF), are also popular methods to explore the long-term temporal information in videos.

Many improvements of video classification are motivated by advances in the image domain. The breakthrough on image classification [20] also rekindled the interest in deep neural networks for video classification. The pioneering work of Karpathy et al. [19] trained 2D-CNN on various forms of stacked video frames from Sports-1M. However, these deep networks are quite inferior to the shallow model based on the best hand-engineered features [44]. This is because complex motions and long-term temporal patterns are difficult to learn only through the simply stacked video frames. Simonyan et al. [36] designed the two-stream networks with two 2D-CNNs on spatial and temporal streams. This method takes advantage of the large-scale ImageNet [20] dataset for pre-training and significantly reduces the complexity to model dynamic motions through optical flow. Ji et al. [17] employed a head tracker and human detector to segment human regions in videos. The segmented regions are stacked as video volumes and used as inputs for 3D-CNN to recognize human actions. Tran et al. [42] applied 3D-CNN on full video frames to avoid pre-processing and jointly captures appearance and motion information. With these methods, similar results to the hand-engineered features [44] have been reported. In contrast to the previous methods with only single or dual modalities, we propose to use four complementary modalities during multimodal fusion.

The aforementioned models only concentrate on motions during short period and lack considerations of long-term temporal clues that are vital for video classification. Several methods have been proposed to address this limitation. Ng et al. [29] explored two schemes to handle full-length videos. They proposed various temporal feature pooling architectures and explored RNN with long short-term memory (LSTM) cells. The trajectory-pooled deep convolutional descriptor (TDD) was presented in [46] to incorporate temporal nature by trajectory constrained sampling and pooling. TDD shares the advantages of both hand-engineered features and deep-learned representations. While the improved networks using RNN can model long-term temporal order, our proposed multimodal method provides multi-temporal scales with short, mid, and long-term time contexts.

Recent work has investigated reasoning across multiple hierarchical levels in a network, which was shown to be advantageous for several tasks. Hariharan et al. [14] proposed hypercolumns for image segmentation and fine-grained localization. A hypercolumn at a given location is the vector containing all the values above that location at all layers of the CNN. DAG-CNN [48] introduced a multi-scale architecture to learn scale-specific features for natural scene recognition. FlowNet [10] preserved feature maps of both coarser and lower layers for optical flow estimation. We propose to extract feature representations from multiple layers to reason at multi-scale abstraction and invariance for video classification.

Combining multiple complementary feature representations is often effective to improve classification. Tamrakar et al. [41] evaluated various early and late fusion strategies in the context of multimedia event detection. Zhang et al. [52] computed non-linear kernels for each feature type and summed up the kernels for SVM training. Multiple kernel learning (MKL) [22] is a popular approach to estimate feature combination weights. However, it was observed [12] that simple averaging and geometric mean were highly competitive to MKL. Jiang et al. [18] proposed to jointly compute a codebook of audio and visual features for video classification with the intention to model correlations between the two modalities. Ngiam et al. [30] proposed a deep autoencoder to enforce cross modality learning between audio and video inputs. Our fusion method differs in combining robust boosting model with deep-learned representations from multiple layers and modalities.

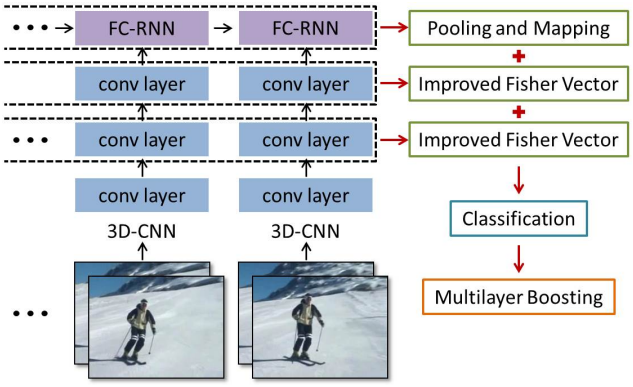

Figure 2: Illustration of multilayer representation and fusion. The proposed feature aggregation methods are used to represent fully connected and convolutional layers over time. The introduced boosting algorithm is applied to combine the representations from multiple layers.

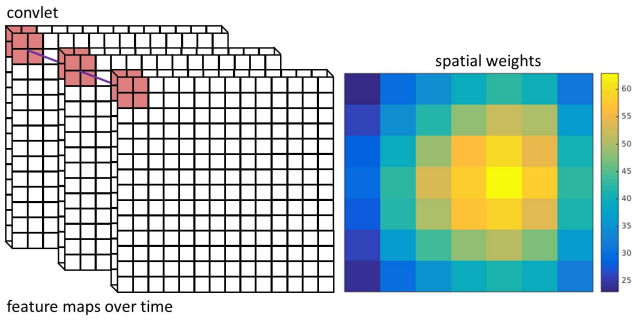

Figure 3: Learning spatial discriminative weights of a convolutional layer by convlets. A spatial weight indicates how discriminative or important that local spatial region is in a convolutional layer.

3. MULTILAYER REPRESENTATIONS

As a hierarchical feed-forward architecture, CNN progressively compute abstract and invariant representations of inputs. Recognition algorithms based on CNN often make predictions based on softmax scores or the last layer which is the summary of variables in the preceding layers. However, we argue that various abstractions such as pose, articulation, parts, objects, etc., learned in the intermediate layers can provide a complete description, from fine-scale to global scale, for video classification. Moreover, we propose a concept of convlet to utilize the spatial information reserved in `conv` layers to refine the final feature representation. In this section, we describe the detailed procedures to compute multilayer representations as illustrated in Fig. 2.

3.1 Improved Fisher Vector with the Convlet

Recent work on visualizing and understanding CNN reveals that `conv` layers demonstrate many intuitively desirable properties such as strong grouping within each feature map and exaggeration of discriminative parts of objects [51]. Therefore, a set of appropriate levels of compositionality in `conv` layers are able to supply plenty of fine-scale information to the category-level semantics. Meanwhile, the features from the layers come for free because they are already extracted during the forward pass. Furthermore, compared to `fc` layers, `conv` layers contain the spatial information, which can be applied to adaptive pooling and feature refinement because the discriminative information for video classification is often unevenly distributed in spatial domain.

We start from defining the convlet which is used to measure the spatial discriminability of activations at a `conv` layer. Assume s_l is the size (height and width) of a feature map and d_l denotes the total number of feature maps. We represent a set of `conv` layers extracted from a video by $\mathcal{C} = \{\boldsymbol{c}_{t,l};\quad t = 1,\ldots,T;\ l = 1,\ldots,L_c\}$, where T is the number of frames or short clips, L_c is the number of selected `conv` layers, and $\boldsymbol{c}_{t,l} \in \mathbb{R}^{s_l \times s_l \times d_l}$ indicates the l-th `conv` layer computed at the t-th timestamp. Since each convolutional kernel can be treated as a latent concept detector [47], we convert $\boldsymbol{c}_{t,l}$ to $s_l \times s_l$ feature descriptors, each of which is with the responses of d_l concept detectors. Thus a video can generate $n_l = s_l \times s_l \times T$ feature descriptors $\boldsymbol{x}_i \in \mathbb{R}^{d_l}$ at the l-th convolutional level, where $i = 1,\ldots,n_l$. Let R indicate the pre-defined spatial neighboring cells over a `conv` layer and R_j denote the j-th cell. We obtain the convlet corresponding to a spatial cell by

$$\boldsymbol{q}_j = \mathcal{G}\left(\{\boldsymbol{x}_i\}_{i \in R_j}\right),\quad j = 1,\ldots,|R|, \tag{1}$$

where $\mathcal{G}$ is a general coding and pooling operator and we employ FV [34] as $\mathcal{G}$ in our experiments. The convlet $\boldsymbol{q}_j$ is a representation that aggregates $\boldsymbol{x}_i$ in a local spatial region across the entire video, as shown in Fig. 3. We then use each convlet $\boldsymbol{q}_j$ to make video classification and the accuracy α_j associated with R_j indicates how discriminative this local spatial cell is in a `conv` layer. We transform the classification accuracy α_j to a spatial discriminative weight w_j with softmax function $w_j = \exp(\alpha_j)/\sum_{k=1}^{|R|}\exp(\alpha_k)$ or sigmoid function $w_j = 1/[1+\exp(\alpha' - \alpha_j)]$, where α' is a parameter to control the relative weight. All features $\boldsymbol{x}_i$ of spatial cell R_j have the same associated weight w_j.

The heat map in Fig. 3 demonstrates the spatial weights learned by convlets of `conv5` in VGG16 [37] on the UCF101 dataset [38]. The features close to boundary regions are much less discriminative than those in the middle, in particular for the left two corner regions. It is also interesting to observe that the hot regions are not exactly centered but a bit shifted towards the right. In addition, spatial weights of different `conv` layers in the same network often exhibit slightly different spatial distributions. Since the weight w_i of $\boldsymbol{x}_i$ represents how discriminative or important $\boldsymbol{x}_i$ is for classification, we can take advantage of this property to improve a general feature aggregation method. We demonstrate the improvement to FV [34] in this paper.

As assumed in FV, the feature descriptors $\boldsymbol{x}_i$ have a Gaussian mixture model (GMM) distribution characterized by parameters $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}$ with $k = 1,\ldots,K$, where π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\sigma}_k$ are the prior mode probability, mean, and covariance (diagonal) of the k-th Gaussian component φ_k. To better fit the diagonal covariance assumption, we apply PCA to decorrelate $\boldsymbol{x}_i$ and reduce feature dimensions. Each feature $\boldsymbol{x}_i$ is then encoded by the deviations with respect to the parameters of GMM. Let $\gamma_{i,k}$ be the soft assignment of $\boldsymbol{x}_i$ to the k-th Gaussian component:

$$\gamma_{i,k} = \frac{\pi_k \varphi_k(\boldsymbol{x}_i)}{\sum_{j=1}^{K}\pi_j \varphi_j(\boldsymbol{x}_i)}. \tag{2}$$

We obtain the improved Fisher vector (iFV) representation of a video at a convolutional layer by concatenating the following derivative vectors from K Gaussian components:

$$\boldsymbol{\rho}_k = \frac{1}{n_l\sqrt{\pi_k}} \sum_{i=1}^{n_l} \gamma_{i,k} w_i \left(\frac{\boldsymbol{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right), \tag{3}$$

$$\boldsymbol{\tau}_k = \frac{1}{n_l\sqrt{2\pi_k}} \sum_{i=1}^{n_l} \gamma_{i,k} w_i \left[\frac{(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right], \tag{4}$$

where $\boldsymbol{\rho}_k$ and $\boldsymbol{\tau}_k$ are the d_l-dimensional derivatives with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ of the k-th Gaussian component. We apply the spatial discriminative factor w_i to weight the relative displacements of $\boldsymbol{x}_i$ to the mean and covariance in Eq. (3–4). In this way, more informative features gain higher contributions to the final representation, while background or noisy features are suppressed. We use iFV to compute the video representations of selected `conv` layers over time.

3.2 Feature Pooling and Mapping

We represent a set of `fc` layers computed from a video by $\mathcal{F} = \{\boldsymbol{f}_{t,l};\ t = 1, \ldots, T;\ l = 1, \ldots, L_f\}$, where $\boldsymbol{f}_{t,l} \in \mathbb{R}^{d_l}$ denotes the l-th `fc` layer computed at timestamp t. The `fc` vector is more sensitive to the category-level semantic information and usually has high dimensions (e.g., $d_l = 4096$ in VGG16). Compared to $\boldsymbol{c}_{t,l}$, which can generate $s_l \times s_l$ features at each timestamp, $\boldsymbol{f}_{t,l}$ is far more sparse as spatial information is lost. Considering these properties, we first apply temporal max pooling to aggregate $\boldsymbol{f}_{t,l}$ across time to obtain $\boldsymbol{f}_l$, which is the initial representation of a video at the l-th `fc` level.

While the last fully-connected layer in a network performs linear classification, it is flexible to inject additional non-linearity to $\boldsymbol{f}_l$ by using non-linear kernels in SVM. However, non-linear SVM is generally much slower than linear one in terms of both learning and prediction. In particular, we are able to train linear SVM in time linear with the number of training samples. This favorably extends the applicability of linear SVM algorithms to large-scale data, which is usually the case for video classification. We thus employ the explicit feature map [43] to approximate large-scale non-linear SVM by the linear one. In explicit feature map, the initial representation $\boldsymbol{f}_l$ is lifted to a Hilbert space with moderately higher feature dimensions through $\psi : \mathbb{R}^{d_l} \rightarrow \mathbb{R}^{d_l(2z+1)}$ such that the inner product in this space can reasonably well approximate a non-linear kernel κ, i.e., $\langle \psi(\boldsymbol{f}_l), \psi(\boldsymbol{f}_l^{'}) \rangle \approx \kappa(\boldsymbol{f}_l, \boldsymbol{f}_l^{'})$. Therefore the final representation $\psi(\boldsymbol{f}_l)$ of a `fc` layer makes use of not only the discriminative power of non-linear kernels but also the efficient training and evaluation of the linear one.

4. FC-RNN STRUCTURE

Most networks hinge on short or mid-term contents such as a single frame [36] or a buffer of frames [42], where features are independently extracted for video classification. We believe that there are important connections between frames of the entire video and that the order of frames matters. To address this intuition, we propose a simple and effective structure, FC-RNN, to transform a network pre-trained on separate frames or clips to deal with video as a whole sequence.

4.1 Initialization of Recurrent Layers

One of the straightforward ways to enable networks to work with video as a sequence is to introduce a stack of recurrent layers on top of the last `fc` layer. This method is common [8, 29] and shows improvement in performance. The output of such a recurrent layer at timestamp t is computed as:

$$\boldsymbol{h}_t = \mathcal{H}(\boldsymbol{W}_{ih}\boldsymbol{f}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + b_h), \tag{5}$$

where $\mathcal{H}$ is an activation function, $\boldsymbol{W}_{ih}$ is the input-to-hidden matrix, $\boldsymbol{f}_t$ is the input to this layer, $\boldsymbol{W}_{hh}$ is the hidden-to-hidden matrix, $\boldsymbol{h}_{t-1}$ is a hidden state vector from previous timestamp, and b_h is an optional bias. Both $\boldsymbol{W}_{ih}$ and $\boldsymbol{W}_{hh}$ are randomly initialized. We refer to this as the standard initialization.

One drawback of the standard initialization is that it requires to train an entire layer (or a stack of layers) from scratch even if a pre-trained network is used for feature extraction. This would result in reducing important generalization properties of a network that is fine-tuned on a relatively small dataset. In this paper, we propose to transform `fc` layers of a pre-trained CNN into recurrent layers. In this way, we preserve the structure of a pre-trained network as much as possible. Assume that a pre-trained `fc` layer at timestamp t has the structure:

$$\boldsymbol{f}_t = \mathcal{H}(\boldsymbol{W}_{io}\boldsymbol{y}_t + b_f), \tag{6}$$

where $\boldsymbol{W}_{io}$ is the pre-trained input-to-output matrix, $\boldsymbol{y}_t$ is output of the previous layer and b_f is bias. We suggest to transform it into a recurrent layer as:

$$\boldsymbol{f}_t = \mathcal{H}(\boldsymbol{W}_{io}\boldsymbol{y}_t + \boldsymbol{W}_{hh}\boldsymbol{f}_{t-1} + b_f). \tag{7}$$

This `fc` initialized recurrent structure is referred as FC-RNN. Fig. 4 illustrates the difference between our proposed FC-RNN and the standard RNN. Our method only introduces a single weight matrix that needs training from scratch, i.e., the hidden-to-hidden matrix $\boldsymbol{W}_{hh}$. Other weight matrices have been already pre-trained and can be just fine-tuned. We observe that this design is effective to reduce over-fitting and expedite convergence. LSTM is not used in our networks because 1) the complicated cell structure in LSTM is not well-adapted to our design; 2) the sequence of clips processed by 3D-CNN in a video is not long as each clip covers a number of non-overlapping frames; 3) LSTM has comparable results to standard RNN in our experiments.

4.2 Regularization

We apply a number of regularization techniques in the training of FC-RNN. The recurrent connection is prone to learn the specific order of videos in the training set, therefore we randomly permute the order of training videos for each epoch. This operation slows down convergence but improves generalization. The regularization term which forces to learn weights with smaller ℓ_2-norm also helps generalization. With intention of preventing the gradients from exploding in recurrent layers, we employ soft gradient clipping in the following way. For each computed gradient g during stochastic gradient descent (SGD), we check if its ℓ_2-norm $\|g\|$ is greater than a pre-defined threshold $\delta = 10$. If that is the case, we rescale the gradient to $g \leftarrow g\delta/\|g\|$. We find that without gradient clipping the explosion of gradient values is a critical barrier to successfully training the networks. To further improve generalization, we train networks with

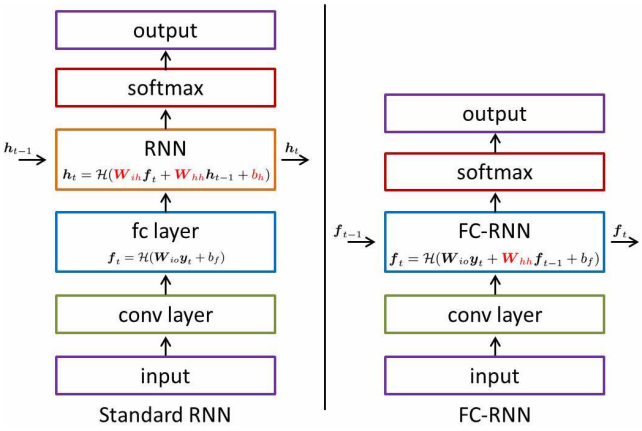

Figure 4: Comparison of standard RNN and FC-RNN. The variables in red correspond to the parameters that need to be trained from scratch.

drop-out on the outputs of recurrent layers. During training, we set the outputs of the recurrent layers to 0 with a probability of $p = 0.5$, and scale the activations of other neurons by a factor of $1/(1-p)$.

5. MULTIMODAL REPRESENTATIONS

Since the visual information in videos is a juxtaposition of not only scenes and objects but also atomic actions evolving over the whole video sequence, it is favorable to capture and combine both static appearances and dynamic motions. To address this challenge we use a multimodal approach to model a variety of semantic clues in multi-temporal scales. Fig. 1 demonstrates our proposed four modalities, which provide mutually complementary information in short, mid, and long-term temporal contexts.

The two networks operating on spatial frames (single frame in 2D-CNN-SF and short clip of frames in 3D-CNN-SF) can capture objects and scenes that are strongly correlated to certain video categories, e.g., snow and mountains in Fig. 1 indicate skiing. 2D-CNN-SF is essentially an image classification network which can be built upon the recent advances in large-scale image recognition methods and datasets. 3D-CNN-SF selectively attends to both motion and appearance cues through spatio-temporal convolution and pooling operations. It encapsulates the mid-term temporal information as the network's input is a short video clip (e.g., 16 spatial frames). We utilize the proposed FC-RNN for 3D-CNN-SF to learn the long-term temporal order. The recurrent structure is not used for 2D-CNN-SF due to very limited improvement (0.1%). This is probably because the static information such as objects and scenes modeled by 2D-CNN-SF is not very correlated with the temporal evolution.

Since optical flow [2] explicitly captures dynamic motions, the two networks running on optical flow images (single image in 2D-CNN-OF and short clip of images in 3D-CNN-OF) provide vital clues to recognize actions. Moreover, optical flow also conveys rough shape cues of moving objects, e.g., the skier and ski poles in Fig. 1. Note, in contrast to the temporal stream [36] used in most previous methods, which work on the stacked optical flow maps, we input a single *colorized optical flow image* to 2D-CNN-OF. As illustrated in Fig. 1, a colorized optical flow image contains 3 channels with RGB values, while an optical flow map includes 2 channels with the raw values of horizontal and vertical displacements. The hue and saturation of an colorized optical flow image indicate flow's orientation and magnitude. This enables us to reduce over-fitting and training time by leveraging pre-trained models from large-scale image datasets. Since the input is a single colorized image, 2D-CNN-OF captures the fine-scale and short-term temporal information between a pair of adjacent frames. 3D-CNN-OF models the high order motion cues such as spatial and temporal derivatives of optical flow, which has been successfully applied to hand-engineered features [44]. This modality also encapsulates the mid-term temporal clues. Similar to 3D-CNN-SF, FC-RNN is also employed to learn the long-term temporal order of 2D-CNN-OF and 3D-CNN-OF.

To obtain the final multimodal representation of a video, we use the aforementioned iFV as well as temporal max pooling and explicit feature map to compute the representations of selected `conv` and `fc` layers (respectively for each modality).

6. FUSION BY BOOSTING

Given the above representations of multiple layers and modalities, in this section, we focus on how to effectively utilize correlations across different representations. We formulate the multilayer and multimodal fusion as a boosting task to maximize the classification accuracy.

We represent a training set by $\{(v_i, y_i)\}_{i=1}^N$ which contains N instance pairs of a video $v_i \in \mathcal{V}$ and a class label $y_i \in \{1, \ldots, C\}$. Let $\{\boldsymbol{r}_m : \mathcal{V} \to \mathbb{R}^{d_m}\}_{m=1}^M$ indicate M video representations extracted by the proposed feature aggregation methods from `conv` and `fc` layers of multiple modalities. We use a general kernel function κ to measure the similarity between instances by the m-th video representation: $\kappa_m(v, v') = \kappa(\boldsymbol{r}_m(v), \boldsymbol{r}_m(v'))$. So the kernel response of a given instance $v \in \mathcal{V}$ to the whole training samples is defined as $\boldsymbol{\mathcal{K}}_m(v) = [\kappa_m(v, v_1), \ldots, \kappa_m(v, v_N)]^T$. We focus on the binary classification problem in the following derivation, which extends straightforwardly to multiple classes. Here the objective is to optimize a linear combination of the predictions using M representations: $U(v) = \sum_{m=1}^M \theta_m u_m(v)$, where θ_m is a mixing coefficient and u_m is a decision function. In this paper, we use SVM with the decision function $u_m(v) = \boldsymbol{\mathcal{K}}_m(v)^T \boldsymbol{a}_m + b_m$, but the weak learner u_m is not necessarily SVM. All parameters of the fusion model can be solved by training u_m based on each individual video representation and subsequently optimizing θ_m through:

$$\begin{aligned} \arg\max_{\theta,\xi,\epsilon} \quad & \epsilon - \frac{1}{\nu N} \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i \sum_{m=1}^M \theta_m u_m(v_i) + \xi_i \geq \epsilon, \quad i = 1, \ldots, N \\ & \sum_{m=1}^M \theta_m = 1, \quad \theta_m \geq 0, \quad m = 1, \ldots, M, \end{aligned} \tag{8}$$

where ξ_i is a slack variable and ν is a regularization parameter to control the smoothness of the resulting function. This is essentially a linear programming problem and can be solved by the column generation approach [7]. Similar to image classification [12], in the multiclass case with C categories we have two variations of the mixing coefficients. We

call the first variant boost-u which jointly learns a uniform coefficient vector $\theta \in \mathbb{R}^M$ for all classes. The alternative one boost-c learns a coefficient vector for each class resulting in a coefficient matrix $\Theta \in \mathbb{R}^{M \times C}$. So the final decision functions for the fusion of multiple layers and modalities with the two boosting variants are:

$$y(v) = \arg\max_{c=1,\ldots,C} \sum_{m=1}^{M} \theta_m \left(\boldsymbol{\mathcal{K}}_m(v)^T \boldsymbol{a}_{c,m} + b_{c,m} \right), \quad (9)$$

$$y(v) = \arg\max_{c=1,\ldots,C} \sum_{m=1}^{M} \Theta_m^c \left(\boldsymbol{\mathcal{K}}_m(v)^T \boldsymbol{a}_{c,m} + b_{c,m} \right). \quad (10)$$

This boosting algorithm is a unified method for both multilayer and multimodal fusion. It can be used by multilayer fusion to combine the video representations $\boldsymbol{r}_m$ from multiple layers in a single modality. If the set of representations is extracted over multiple modalities, it then performs as multimodal fusion. We observe that the joint fusion of multiple layers over all modalities is slightly better than the separate fusion of individual modality first then across all modalities. This is probably because the joint fusion allows different modalities to explore better correlations at different levels.

7. EXPERIMENTS

In this section, we extensively evaluate the proposed multilayer and multimodal fusion method on two public benchmark datasets for video classification: UCF101 [38] and HMDB51 [21]. In all experiments, we use LIBLINEAR [9] as the linear SVM solver. Experimental results show that our algorithm achieves the state-of-the-art results on the two benchmarks.

7.1 Experimental Setup

7.1.1 Datasets

The UCF101 [38] dataset contains 101 action classes with large variations in scale, viewpoint, illumination, camera motion, and cluttered background. It consists of 13,320 videos in total. We follow the standard experimental setting as in [38] and use three training and testing splits. In each split, 20% of training data is randomly selected as validation set for the boosting model selection. The first split of UCF101 (denoted as UCF101*) is also used to evaluate and understand the contribution of each individual component. We report the average accuracy over the three splits as the overall measurement.

The HMDB51 dataset [21] is collected from a wide range of sources from digitized movies to online videos. It contains 51 categories and 6,766 videos in total. This dataset includes original videos and stabilized ones. Our evaluations are based on the original version. There are 70 videos for training and 30 videos for testing in each class. We use 40% of training data as validation set to perform model selection for boosting. We follow the evaluation protocol defined in [21] and use three training and testing splits and report the mean accuracy over the three splits.

7.1.2 Implementations

We implement the networks of four modalities in Theano with cuDNN4 on an NVIDIA DIGITS DevBox with four Titan X GPUs. 2D-CNN and 3D-CNN in the experiments are initialized with VGG16 [37] pre-trained on ImageNet and C3D [42] pre-trained on Sports-1M, respectively. Outputs of the last four layers of each network are used to represent videos. Special attention is paid to assembling mini-batches in order to deal with varying video length. We fill all frames of a video into a mini-batch and use another video if there is still space in the mini-batch. When the limit of a mini-batch is reached and there are frames left, we use them in the next one. When there are no more frames to fill a mini-batch, we fill it with zeros and these examples are not used in computation. We shuffle video instances after each epoch to prevent learning a specific sequence of examples. The last hidden state vector of each mini-batch is propagated to the next batch.

We apply data augmentations to increase the diversity of videos. For 2D-CNN we skip every second frame and operate on a single frame resized to 320×240 and cropped to 224×224. 3D-CNN works on a clip of 16 frames resized to 160×120 and cropped to 112×112. Training frames are generated by random cropping and flipping video frames, while for testing, only a central crop with no flipping is evaluated. Since the two datasets are of quite different sizes, we apply different learning rate schedules. For UCF101, we fine-tune 9 epochs with an initial learning rate of $\lambda = 3 \times 10^{-4}$ and divide it by 10 after each 4 epochs. For HMDB51, we perform fine-tuning for 30 epochs with the same initial learning rate and divide it by 10 after every 10 epochs. All network parameters that do not have pre-trained weights are initialized with random samples drawn from a zero-mean normal distribution ($\sigma = 0.01$).We use the frame-wise negative log-likelihood of a mini-batch as the cost function, which is optimized using SGD with a momentum of 0.9.

7.2 Experimental Results

7.2.1 Evaluation of Feature Aggregations

We first evaluate the performance of iFV to represent `conv` layers in different modalities. Compared to the traditional aggregation methods, iFV retains high-order statistics; in particular, it adaptively weights the features of a `conv` layer according to the associated spatial weights learned by the proposed convlet. We keep 300 out of 512 components in PCA. For computing the spatial discriminative weights, we find the sigmoid is more discriminative than softmax, e.g., iFV with sigmoid outperforms that with softmax by 0.6% for `conv5` layer in 2D-CNN-SF. The sigmoid function is therefore used in the following experiments. We set $K = 128$ Gaussian components for both methods so the final feature dimensionality is 76.8K. We compare iFV with the conventional FV [34] in Table 1, where iFV consistently outperforms FV for `conv` layers in all modalities with the improvements ranging from 0.6% to 2.5%. A larger improvement is observed for `conv4` than `conv5`, probably because of the finer spatial information preserved in the lower layer. These improvements clearly show the advantages of utilizing the spatial discriminability learned by convlets to enhance the feature representation.

We employ temporal max pooling to aggregate `fc` layers, which are further extended with an explicit feature map to approximate non-linear kernels. This representation also benefits from the same efficiency of learning and prediction as linear SVM. We demonstrate the results of `fc` layers in 3D-CNN-SF with approximated non-linearities in Table 2.

Modality	Layer	FV [34]	iFV
2D-CNN-SF	conv4	74.2%	**76.7%**
	conv5	79.6%	**80.6%**
2D-CNN-OF	conv4	75.6%	**78.1%**
	conv5	81.9%	**82.6%**
3D-CNN-SF	conv4	83.6%	**84.8%**
	conv5	83.3%	**84.6%**
3D-CNN-OF	conv4	78.2%	**78.8%**
	conv5	78.1%	**78.7%**

Table 1: Comparison of FV and the proposed iFV to represent convolutional layers of different modalities on UCF101*.

Layer	Linear	χ^2	Jensen-Shannon	Intersection
fc6	84.1%	84.8%	84.6%	**84.9%**
fc7	82.4%	82.9%	83.0%	**83.2%**

Table 2: Comparison of different non-linear approximations to represent fully connected layers in 3D-CNN-SF on UCF101*.

Both fc6 and fc7 are transformed to recurrent layers by FC-RNN. We use the ℓ_2-norm and $z = 3$ in the explicit feature map, so the extended feature dimension is 28,672. The baseline method is the linear representation by temporal max pooling without feature mapping. We evaluate three additive non-linear kernels: χ^2, Jensen-Shannon and intersection kernels, which are widely used in machine learning and computer vision. All non-linear representations outperform the linear one, especially the representation with intersection kernel achieves the best results. We thus use the intersection non-linearity approximation to represent fc layers in the following experiments.

7.2.2 *Evaluation of FC-RNN*

Our method extracts static and dynamic information at multiple temporal scales. 2D-CNN and 3D-CNN on spatial frames and optical flow images compute features from short-term and mid-term temporal contexts. FC-RNN is then employed to model each video as an ordered sequence of frames or clips to capture the long-term temporal order. Since FC-RNN maintains the structure of a pre-trained network to the greatest extent, it is therefore effective at preserving important generalization properties of the network, when fine-tuned on a smaller target dataset. Moreover, FC-RNN achieves higher accuracy and is faster to converge compared to the standard RNN. We compare the training and testing performances of our proposed FC-RNN and the standard RNN in Fig. 5. To avoid clutter, we only show this comparison for 3D-CNN modalities—a similar phenomena is observed on 2D-CNN-OF as well. FC-RNN is generally able to alleviate over-fitting and converge faster, e.g., FC-RNN outperforms standard RNN and LSTM by 3.0% and 2.9% on 3D-CNN-SF. In comparison to the networks without recurrent connections, FC-RNN significantly improves the modalities of 2D-CNN-OF, 3D-CNN-SF and 3D-CNN-OF by 3.3%, 3.2% and 5.1%, respectively. This demonstrates the benefits of FC-RNN in modeling the long-term temporal clues.

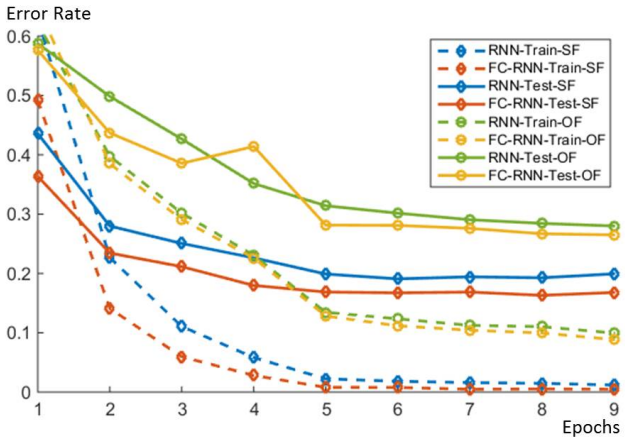

Figure 5: Comparison of the proposed FC-RNN and the standard RNN in training and testing of 3D-CNN-SF and 3D-CNN-OF on UCF101*.

Initialized by 3D-CNN-SF			✓	✓
Using FC-RNN		✓		✓
Accuracy of 3D-CNN-OF	68.4%	70.4%	72.5%	**73.5%**

Table 4: Comparison of the initialization methods for 3D-CNN-OF on UCF101*.

7.2.3 *Evaluation of Multilayer Fusion*

Here we evaluate the multilayer fusion on combining various layers for individual modalities. Table 3 shows the performance of each single layer across different modalities and the fusion results on the two datasets. Although the last layer in a network is the most sensitive to category-level semantics, it is not unusual for lower layers to have on par or superior results, e.g., conv5 of 2D-CNN-OF on UCF101 and conv5 of 2D-CNN-SF on HMDB51. So it is of great potential to exploit the intermediate abstractions such as parts, objects, poses, articulations and so on for video classification. It is also of interest to observe that most layers produce accuracies better than the baseline of softmax, i.e., the prediction outputs of a network. This again validates the merit of the proposed feature aggregation methods to represent conv and fc layers.

If we use the boosting algorithm to combine multiple layers, the fusion result significantly outperforms the baseline for all modalities, especially for 3D-CNN-OF with 7.2% and 7.9% gains on UCF101 and HMDB51. This demonstrates that various abstractions extracted in multiple layers are of rich complementarity. Although boost-c is more flexible to have class-specific mixing coefficients, its results are inferior to those of boost-u. This is because the model of boost-c tends to over-fit, since the $C \times M$ parameters to fit in boost-c require more training data than the M parameters in boost-u. We thus use boost-u in the following fusion experiments. 3D-CNN-SF is the best modality before fusion as it jointly models appearance and motion information. After multilayer fusion the other two modalities involving dynamic cues are enhanced to a similar performance level, which shows that the boosting method successfully maximizes the capability of a network.

	UCF101 (%)				HMDB51 (%)			
	2D-CNN-SF	2D-CNN-OF	3D-CNN-SF	3D-CNN-OF	2D-CNN-SF	2D-CNN-OF	3D-CNN-SF	3D-CNN-OF
`conv4`	75.0	79.7	83.1	80.2	37.0	41.7	49.1	48.9
`conv5`	79.9	83.9	83.7	80.6	42.0	47.0	49.9	48.6
`fc6`	81.0	81.1	84.0	80.3	42.7	48.3	51.8	51.5
`fc7`	80.5	82.7	83.7	79.9	41.6	47.3	52.1	50.0
softmax	79.5	80.9	82.9	75.3	40.2	47.9	51.2	45.1
boost-c	82.1(+2.6)	84.4(+3.5)	84.5(+1.6)	81.4(+6.1)	43.8(+3.6)	50.6(+2.7)	52.2(+1.0)	52.8(+7.7)
boost-u	**82.6**(+3.1)	**85.9**(+5.0)	**85.4**(+2.5)	**82.5**(+7.2)	**44.5**(+4.3)	**51.4**(+3.5)	**53.1**(+1.9)	**53.0**(+7.9)

Table 3: Performances of individual layers over different modalities and multilayer fusion results.

Modality	Accuracy	Combinations						
2D-CNN-SF	83.2	✓		✓	✓	✓		✓
2D-CNN-OF	84.8	✓	✓			✓	✓	✓
3D-CNN-SF	85.9		✓	✓		✓	✓	✓
3D-CNN-OF	81.4				✓		✓	✓
Fusion Accuracy		90.3	90.8	87.1	90.4	91.2	91.3	**91.9**

Table 5: Classification accuracies (%) of different modalities and various combinations on UCF101*.

7.2.4 Evaluation of Multimodal Fusion

We now demonstrate multimodal fusion, combining the proposed four modalities. Our networks are initialized by models pre-trained on large-scale image and video datasets so it is natural to fine-tune these networks for the two modalities of spatial frames. However, for the other two modalities involving optical flow, they are distant from the source if we regard fine-tuning as a way of domain transformation. We introduce a simple but effective method to bridge the two domains—initialize optical flow networks by spatial frame models that have been fine-tuned on the target domain. As shown in Table 4, compared to the networks directly fine-tuned on the source model (i.e., not initialized by 3D-CNN-SF), our initialization remarkably improves the results.

Table 5 contains the accuracies for various combinations of the four modalities. Observe that fusing any pair of modalities improves the individual results. The best classification accuracy of 91.9% is obtained by the combination of all modalities. In the end, we achieve an accuracy of 91.6% on UCF101 and 61.8% on HMDB51 through combining the four modalites by boost-u. In comparison to the results in Table 3, the multimodal fusion produces much higher accuracy than any individual modality. This indicates the strong complementarity between the four modalities that capture diverse static and dynamic features at multiple temporal scales. In comparison to the baseline fusion methods, boost-u improves the result by 2.3% over geometric mean [41], 4.3% over SVM-based fusion [36], and 7.9% over AdaBoost [5] on UCF101*. This demonstrates that boost-u is more effective to exploit and fuse the complementary relationship of multiple modalities.

We finally compare our results with the most recent state-of-the-art methods in Table 6. Our method produces the best accuracy on UCF101 with a clear margin over other competing algorithms. It is more challenging to fine-tune networks and train boost-u on HMDB51, where each training split is 2.6 times smaller than UCF101. Our method still achieves superior performance on HMDB51. Other competitive results [33, 46] are based on the improved dense trajectories, which require quite a few hand-crafted processes such as dense point tracking, human detection, camera motion estimation, etc. As shown on UCF101, large training data is beneficial for training networks and boosting, so we are planing to explore techniques such as multi-task learning and temporal elastic deformation to increase the effective training size of HMDB51.

UCF101 (%)		HMDB51 (%)	
STIP + BOVW [21]	43.9	STIP + BOVW [21]	23.0
DT + MVSV [4]	83.5	DT + MVSV [4]	55.9
iDT + HSV [33]	87.9	iDT + HSV [33]	61.1
C3D [42]	85.2	iDT + FV [44]	57.2
LRCN [8]	82.9	Motionlets [3]	42.1
TDD [46]	90.3	TDD [46]	**63.2**
RNN-FV [25]	88.0	RNN-FV [25]	54.3
Two-Stream [36]	88.0	Two-Stream [36]	59.4
MultiSource CNN [32]	89.1	MultiSource CNN [32]	54.9
Composite LSTM [39]	84.3	Composite LSTM [39]	44.1
Ours	**91.6**	Ours	**61.8**

Table 6: Comparison of the multimodal fusion to the state-of-the-art results.

8. CONCLUSION

In this paper, we have presented a novel framework to fuse deep neural networks in multiple layers and modalities for video classification. A multilayer strategy is proposed to incorporate various levels of semantics in each single network. We employ effective feature aggregation methods, i.e., iFV and explicit feature maps to represent `conv` and `fc` layers. We further introduce a multimodal approach to capture diverse static and dynamic cues from four highly complementary modalities at multiple temporal scales. FC-RNN is then proposed to effectively model long-term temporal order by leveraging the generalization properties of pre-trained networks. A powerful boosting model is used for the optimal combination of multilayer and multimodal representations. We evaluate our approach extensively on two public benchmark datasets and achieve superior results compared to a number of recent methods.

9. REFERENCES

[1] D. Borth, T. Chen, R. Ji, and S. Chang. Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In *ACM Multimedia*, 2013.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[3] Z. Cai, L. Wang, X. Peng, and Y. Qiao. Motionlets: mid-level 3D parts for human motion recognition. In *CVPR*, 2013.

[4] Z. Cai, L. Wang, X. Peng, and Y. Qiao. Multi-view super vector for action recognition. In *CVPR*, 2014.

[5] S. Chen, J. Wang, Y. Liu, C. Xu, and H. Lu. Fast feature selection and training for AdaBoost-based concept detection with large scale datasets. In *ACM Multimedia*, 2010.

[6] G. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *TASLP*, 2012.

[7] A. Demiriz, K. Bennett, and J. Taylor. Linear programming boosting via column generation. *JMLR*, 2000.

[8] J. Donahue, L. Hendricks, S. Guadarrama, and M. Rohrbach. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[9] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 2008.

[10] P. Fischer, A. Dosovitskiy, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and T. Brox. FlowNet: learning optical flow with convolutional networks. In *ICCV*, 2015.

[11] J. Francoise, N. Schnell, and F. Bevilacqua. A multimodal probabilistic model for gesture based control of sound synthesis. In *ACM Multimedia*, 2011.

[12] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

[13] J. Gemert, C. Veenman, A. Smeulders, and J. Geusebroek. Visual word ambiguity. *TPAMI*, 2009.

[14] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CoRR*, 2015.

[16] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[17] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *TPAMI*, 2013.

[18] W. Jiang, C. Cotton, S.-F. Chang, D. Ellis, and A. Loui. Short-term audio-visual atoms for generic video concept classification. In *ACM Multimedia*, 2009.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[21] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[22] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 2004.

[23] I. Laptev. On space-time interest points. *IJCV*, 2005.

[24] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[25] G. Lev, G. Sadeh, B. Klein, and L. Wolf. RNN Fisher vectors for action recognition and image annotation. In *CoRR*, 2015.

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[27] D. Lowe. Scale invariant feature transform. *IJCV*, 2004.

[28] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[29] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: deep networks for video classification. In *CVPR*, 2015.

[30] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *ICML*, 2011.

[31] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 2001.

[32] E. Park, X. Han, T. Berg, and A. Berg. Combining multiple sources of knowledge in deep CNNs for action recognition. In *WACV*, 2016.

[33] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. *CVIU*, 2016.

[34] F. Perronnin, J. Sanchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale visual recognition. In *ICLR*, 2015.

[38] K. Soomro, A. Zamir, and M. Shah. UCF101: a dataset of 101 human actions classes from videos in the wild. In *CoRR*, 2012.

[39] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

[40] K. Tai, R. Socher, and C. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[41] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.

[42] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. In *ICCV*, 2015.

[43] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *TPAMI*, 2012.

[44] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *IJCV*, 2015.

[45] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[46] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[47] Z. Xu, Y. Yang, and A. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, 2015.

[48] S. Yang and D. Ramanan. Multi-scale recognition with DAG-CNNs. In *ICCV*, 2015.

[49] X. Yang, Z. Liu, E. Zavesky, D. Gibbon, B. Shahraray, and Y. Tian. AT&T Research at TRECVID 2013: surveillance event detection. In *NIST TRECVID Workshop*, 2013.

[50] X. Yang and Y. Tian. Action recognition using super sparse coding vector with spatio-temporal awareness. In *ECCV*, 2014.

[51] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[52] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 2007.