Exploiting Image-trained CNN Architectures for Unconstrained Video Classification

Shengxin Zha[1]
szha@u.northwestern.edu
Florian Luisier[2]
fluisier@bbn.com
Walter Andrews[2]
wandrews@bbn.com
Nitish Srivastava[3]
nitish@cs.toronto.edu
Ruslan Salakhutdinov[3]
rsalakhu@cs.toronto.edu

[1] Northwestern University
Evanston IL USA
[2] Raytheon BBN Technologies
Cambridge, MA USA
[3] University of Toronto
Toronto, Ontario, Canada

Abstract

We conduct an in-depth exploration of different strategies for doing event detection in videos using convolutional neural networks (CNNs) trained for image classification. We study different ways of performing spatial and temporal pooling, feature normalization, choice of CNN layers as well as choice of classifiers. Making judicious choices along these dimensions led to a very significant increase in performance over more naive approaches that have been used till now. We evaluate our approach on the challenging TRECVID MED'14 dataset with two popular CNN architectures pretrained on ImageNet. On this MED'14 dataset, our methods, based entirely on image-trained CNN features, can outperform several state-of-the-art non-CNN models. Our proposed late fusion of CNN- and motion-based features can further increase the mean average precision (mAP) on MED'14 from 34.95% to 38.74%. The fusion approach yields 89.6% classification accuracy on the challenging UCF-101 dataset.

1 Introduction

The huge volume of videos that are nowadays routinely produced by consumer cameras and shared across the web calls for effective video classification and retrieval approaches. One straightforward approach considers a video as a set of images and relies on techniques designed for image classification. This standard image classification pipeline consists of three processing steps: first, extracting multiple carefully engineered local feature descriptors (e.g., SIFT [22] or SURF [2]); second, the local feature descriptors are encoded using the bag-of-words (BoW) [4] or Fisher vector (FV) [26] representation; and finally, classifier is trained (e.g., support vector machines (SVMs)) on top of the encoded features. The main limitation of directly employing the standard image classification approach to video is the lack of exploitation of motion information. This shortcoming has been addressed by extracting optical flow based descriptors (eg. [21]), descriptors from spatiotemporal interest points (e.g., [6, 17, 18, 35, 38]) or along estimated motion trajectories (e.g., [11, 13, 34, 36, 37]).

Pages 60.1-60.13
DOI: https://dx.doi.org/10.5244/C.29.60

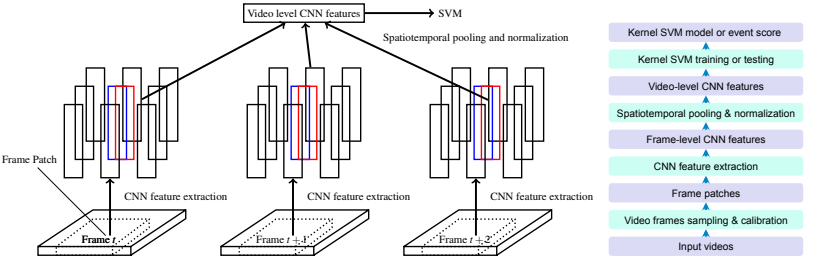


Figure 1: Overview of the proposed video classification pipeline.

The success of convolutional neural networks (CNNs) [20] on the ImageNet [7] has attracted considerable attentions in the computer vision research community. Since the publication of the winning model [15] of the ImageNet 2012 challenge, CNN-based approaches have been shown to achieve state-of-the-art on many challenging image datasets. For instance, the trained model in [15] has been successfully transferred to the PASCAL VOC dataset and used as a mid-level image representation [8]. They showed promising results in object and action classification and localization [24]. An evaluation of off-the-shelf CNN features applied to visual classification and visual instance retrieval has been conducted on several image datasets [28]. The main observation was that the CNN-based approaches outperformed the approaches based on the most successful hand-designed features.

Recently, CNN architectures trained on videos have emerged, with the objective of capturing and encoding motion information. The 3D CNN model proposed in [12] outperformed baseline methods in human action recognition. The two-stream convolutional network proposed in [29] combined a CNN model trained on appearance frames with a CNN model trained on stacked optical flow features to match the performance of hand-crafted spatiotemporal features. The CNN and long short term memory (LSTM) model have been utilized in [23] to obtain video-level representation.

In this paper, we propose an efficient approach to exploit off-the-shelf *image-trained* CNN architectures for video classification (Figure 1). Our contributions are the following:

- We discuss each step of the proposed video classification pipeline, including the choice of CNN layers, the video frame sampling and calibration, the spatial and temporal pooling, the feature normalization, and the choice of classifier. All our design choices are supported by extensive experiments on the TRECVID MED'14 video dataset.
- We provide thorough experimental comparisons between the CNN-based approach and some state-of-the-art static and motion-based approaches, showing that the CNN-based approach can outperform the latter, both in terms of accuracy and speed.
- We show that integrating motion information with a simple average fusion considerably improves classification performance, achieving the state-of-the-art performance on TRECVID MED'14 and UCF-101.

Our work is closely related to other research efforts towards the efficient use of CNN for video classification. While it is now clear that CNN-based approaches outperform most state-of-the-art handcrafted features for image classification [28], it is not yet obvious that this holds true for video classification. Moreover, there seems to be mixed conclusions regarding the benefit of training a spatiotemporal vs. applying an image-trained CNN architecture on videos. Indeed, while Ji *et al.* [12] observed a significant gain using 3D convolutions over the 2D CNN architectures and Simonyan *et al.* [29] obtained substantial gains over an appearance based 2D CNN using optical flow features alone, Karpathy *et al.* [14] reported

only a moderate improvement. Although the specificity of the considered video datasets might play a role, the way the 2D CNN architecture is exploited for video classification is certainly the main reason behind these contradictory observations. The additional computational cost of training on videos is also an element that should be taken into account when comparing the two options. Prior to training a spatiotemporal CNN architecture, it thus seems legitimate to fully exploit the potential of image-trained CNN architectures. Obtained on a highly heterogeneous video dataset, we believe that our results can serve as a strong 2D CNN baseline against which to compare CNN architectures specifically trained on videos.

2 Deep Convolutional Neural Networks

Convolutional neural networks [20] consist of layers of spatially-structured hidden units. Each hidden unit typically looks at a small patch of hidden (or input) units in the previous layer, applies convolution or pooling operations to it and then non-linearity to the result to compute its own state. A spatial patch of units is convolved with multiple filters (learned weights) to generate feature maps. A pooling operation takes a spatial patch and computes typically maximum or average activation (max or average pooling) of that patch for each input channel, creating translation invariance. At the top of the stacked layers, the spatially organized hidden units are usually densely connected, which eventually connect to the output units (e.g. softmax for classification). Regularizers, such as ℓ_2 decay and dropout [32], have been shown to be effective for preventing overfitting. The structure of deep neural nets enables the hidden layers to learn rich distributed representations of the input images. The densely connected layers, in particular, can be seen as learning a high-level representation of the image accumulating information from all the spatial locations.

We initially developed our work on the CNN architecture by Krizhevsky *et al.* [15]. In the recent ILSVRC-2014 competition, the top performing models GoogLeNet [33] and VGG [30] used smaller receptive fields and increased depth, showing superior performance over [15]. We adopt the publicly available pretrained VGG model for its superior post-competition single model performance over GoogLeNet and the popular Krizhevsky's model in our system. The proposed video classification approach is generic with respect to the CNN architectures, therefore, can be adapted to other CNN architectures as well.

3 Video Classification Pipeline

Figure 1 gives an overview of the proposed video classification pipeline. Each component of the pipeline is discussed in this sections.

Choice of CNN Layer We have considered the output layer and the last two hidden layers (fully-connected layers) as CNN-based features. The 1,000-dimensional output-layer features, with values between $[0,1]$, are the posterior probability scores corresponding to the 1,000 classes from the ImageNet dataset. Since our events of interest are different from the 1,000 classes, the output-layer features are rather sparse (see Figure 2 left panel). The hidden-layer features are treated as high-level image representations in our video classification pipeline. These features are outputs of rectified linear units (RELUs). Therefore, they are lower bounded by zero but do not have a pre-defined upper bound (see Figure 2 middle and right panels) and thus require some normalization.

Video Frame Sampling and Calibration We uniformly sampled 50 to 120 frames depending on the clip length. We have explored alternative frame sampling schemes (e.g.,

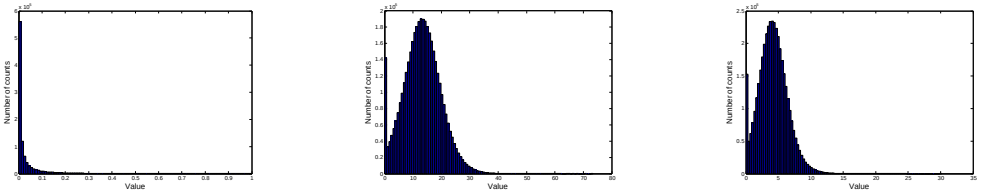
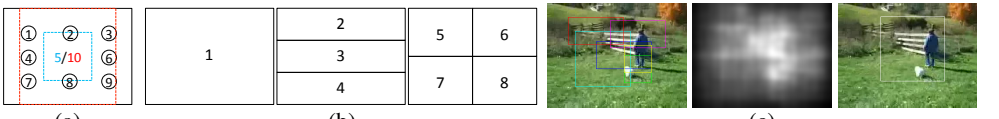


Figure 2: Distribution of values for the output layer and the last two hidden layers of the CNN architecture. Left: output. Middle: hidden6. Right: hidden7.



(a) (b) (c)

Figure 3: Pooling. (a). The 10 overlapping square patches extracted from each frame for spatial pyramid pooling. The red patch (centered at location 10) has sides equal to the frame height, whereas the other patches (centered at locations 1-9) have sides equal to half of the frame height. (b). Spatial pyramids (SP). Left: the 1×1 spatial partition includes the entire frame. Middle: the 3×1 spatial partitions include the top, middle and bottom of the frame. Right: the 2×2 spatial partitions include the upper-left, upper-right, bottom-left and bottom-right of the frame. (c). Objectness-based pooling. Left: objectness proposals. Middle: sum of 1000 objectness proposals. Right: foreground patch from thresholding proposal sums.

based on keyframe detection), but found that they all essentially yield the same performance as uniform sampling. The first layer of the CNN architecture takes 224×224 RGB images as inputs. We extracted multiple patches from the frames and rescale each of them to a size of 224×224. The number, location and original size of the patches are determined by selected spatial pooling strategies. Each of P patches is then used as input to the CNN architecture, providing P different outputs for each video frame.

Spatial and Temporal Pooling We evaluated two pooling schemes, average and max, for both spatial and temporal pooling along with two spatial partition schemes. Inspired by the *spatial pyramid* approach developed for bags of features [19], we spatially pooled together the CNN features computed on patches centered in the same pre-defined sub-region of the video frames. As illustrated in Figure 3, 10 overlapping square patches are extracted from each frame. We have considered 8 different regions consisting of 1×1, 3×1 and 2×2 partitions of the video frame. For instance, features from patches 1, 2, 3 will be pooled together to yield a single feature in region 2. Such region specification considers the full frame, the vertical structure and four corners of a frame. The pooled features from 8 spatial regions are concatenated into one feature vector. Our spatial pyramid approach differs from that of [9, 10, 39] in that it is applied to the output- or hidden-layer feature response of multiple inputs rather than between the convolutional layer and fully-connected layer. The pre-specified regions are also different. We have also considered utilizing *objectness* to guide feature pooling by concatenating the CNN features extracted from the foreground region and the full frame. As shown in Figure 3, the foreground region is resulted from thresholding the sum of 1000 objectness proposals generated from BING [3]. Unlike R-CNN [8] that extract CNN features from all objectness proposals for object detection and localization, We used only one coarse foreground region to reduce the computational cost.

Feature Normalization We compared different normalization schemes and identified normalization most appropriate for the particular feature type. Let $\mathbf{f} \in \mathbb{R}^D$ denote the original video-level CNN feature vector and $\tilde{\mathbf{f}} \in \mathbb{R}^D$ its normalized version. We have investigated three different normalizations: ℓ_1 normalization $\tilde{\mathbf{f}} = \mathbf{f}/\|\mathbf{f}\|_1$; ℓ_2 normalization $\tilde{\mathbf{f}} = \mathbf{f}/\|\mathbf{f}\|_2$, which is typically performed prior to training an SVM model; and *root* normalization $\tilde{\mathbf{f}} = \sqrt{\mathbf{f}/\|\mathbf{f}\|_1}$ introduced in [1] and shown to improve the performance of SIFT descriptors.

Choice of Classifier We applied SVM classifiers to the video-level features, including linear SVM and non-linear SVM with Gaussian radial basis function (RBF) kernel $\exp\{-\gamma||\mathbf{x}-\mathbf{y}||^2\}$ and exponential χ^2 kernel $\exp\{-\gamma\sum_i \frac{(x_i-y_i)^2}{x_i+y_i}\}$. Principal component analysis (PCA) can be applied prior to SVM to reduce feature dimensions.

4 Modality Fusion

4.1 Fisher Vectors

An effective approach to video classification is to extract multiple low-level feature descriptors; and then encode them as a fixed-length video-level Fisher vector (FV). The FV is a generalization of the bag-of-words approach that encodes the zero-order, the first- and the second-order statistics of the descriptors distribution. As low-level features, we considered both the standard D-SIFT descriptors [22] and the more sophisticated motion-based improved dense trajectories (IDT) [34]. For the SIFT descriptors, we opted for multiscale (5 scales) and dense (stride of 4 pixels in both spatial dimensions) sampling, root normalization and spatiotemporal pyramid pooling. For the IDT descriptors, we concatenated HOG [6], HOF [6] and MBH [18] descriptors extracted along the estimated motion trajectory.

4.2 Fusion

We investigated modality fusion to fully utilize the CNN features and the strong hand-engineered features. The weighted average fusion was applied: first, the SVM margins was converted into posterior probability scores using Platt's method [27]; and then the weights of the linear combination of the considered features scores for each class was optimized using cross-validation. We evaluated fusion features scores of different CNN layers and FVs.

5 Evaluation in Event Detection

5.1 Video Dataset and Performance Metric

We conducted extensive experiments on the TRECVID multimedia event detection (MED)'14 video dataset [25]. This dataset consists of: (a) a training set of 4,992 unlabeled *background* videos used as the negative examples; (b) a training set of 2,991 *positive* and *near-miss* videos including 100 positive videos and about 50 near-miss videos (treated as negative examples) for each of the 20 pre-specified events; and (c) a test set of 23,953 videos contains positive and negative instances of the pre-specified events. Some sample frames are given in Figure 4. Contrary to other popular video datasets, such as UCF-101 [31], the MED'14 dataset is not constrained to any class of videos. It consists of a heterogeneous set of temporally untrimmed YouTube-like videos of various resolutions, quality, camera motions, and illumination conditions. This dataset is thus one of the largest and the most challenging dataset for video event detection. As a retrieval performance metric, we considered the one used in the official MED'14 task, i.e., mean average precision (mAP) across all events. The mAP is normalized between 0 (low classification performance) and 1 (high classification performance). In this paper, we will report it in percentage value.

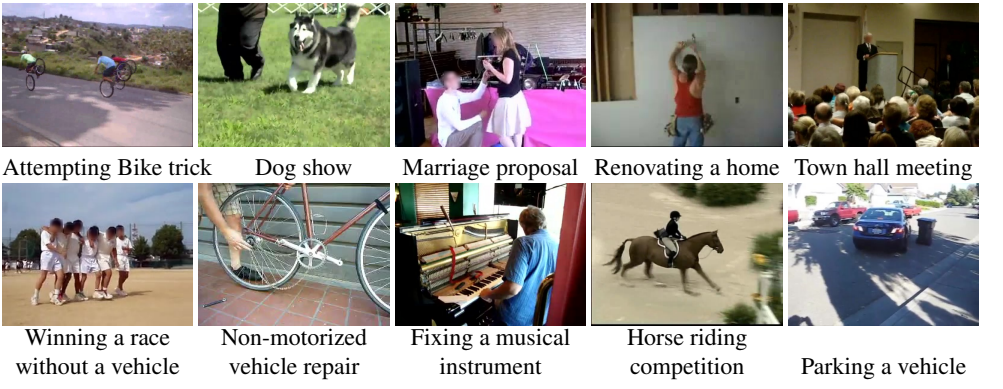


Figure 4: Sample frames from the TRECVID MED '14 dataset.

5.2 Single Feature Performance

The single feature performance with various configurations are reported in Table 1 and 2. The following are a few salient observations.

CNN architectures and layers The deeper CNN architecture yields consistently better performance resulted from the depths and small receptive field in all convolutional layers. We also observed that both hidden layers outperformed the output layer when the CNN architecture, normalization and spatiotemporal pooling strategies are the same.

Pooling, spatial pyramids and objectness We observed a consistent gain for max pooling over average pooling for both spatial and temporal pooling, irrespectively of the used CNN layer. It is mainly resulted from the highly heterogeneous structure of the video dataset. A lot of videos contain frames that can be considered irrelevant or at least less relevant than others; e.g., introductory text and black frames. Hence it is beneficial to use the maximum features response, instead of giving an equal weight to all features. As observed in Table 1, concatenating the 8 spatial partitions (SP8) gives the best performance in all CNN layers and SVM choices (up to 6% mAP gain over no spatial pooling, i.e., "none"), but at the expense of an increased feature dimensionality and consequently, an increased training and testing time. Alternatively, objectness-guided pooling provides a good trade-off between performance and dimensionality, shown in Table 2. It outperforms the baseline approach without spatial expansion ("none" in Table 1). The feature dimensions are only one-fourth of that of SP8; while the performance nearly matches that of SP8 using kernel SVM.

Normalization We observed that ℓ_1 normalization did not perform well; while the ℓ_2 normalization is essential to achieve good performance with hidden layers. Applying root normalization to the output layer yields essentially the same result as applying ℓ_2. Yet, we noticed a drop in performance when the root normalization was applied to the hidden layers.

Classifier One SVM model was trained for each event following the TRECVID MED'14 training rules; i.e., excluding the positive and near-miss examples of the other events. We observed that kernel SVM consistently outperformed linear SVM regardless of CNN layer or normalization. For the output layer, which essentially behaves like a histogram-based feature, the χ^2 kernel yields essentially the same result as RBF kernel. For the hidden layers, the best performance was obtained with a RBF kernel. As shown in Table 1, the result of kernel SVM largely outperforms that of linear SVM.

PCA We analyzed dimensionality reduction of the top performing feature, CNN-B-

Layer	Dim.	SP	Norm	SVM	mAP A	mAP B
output	1,000	none	root	linear	15.90%	19.46%
output	8,000	SP8	root	linear	22.04%	25.67%
output	8,000	SP8	ℓ_2	linear	22.01%	25.88%
output	1,000	none	root	χ^2	21.54%	25.30%
output	8,000	SP8	root	χ^2	27.22%	31.24%
output	8,000	SP8	root	RBF	27.12%	31.73%
hidden6	4,096	none	ℓ_2	linear	22.11%	25.37%
hidden6	32,768	SP8	root	linear	21.13%	26.27%
hidden6	32,768	SP8	ℓ_2	linear	23.21%	28.31%
hidden6	4,096	none	ℓ_2	RBF	28.02%	32.93%
hidden6	32,768	SP8	ℓ_2	RBF	28.20%	33.54%
hidden7	4,096	none	ℓ_2	linear	21.45%	25.08%
hidden7	32,768	SP8	root	linear	23.72%	26.57%
hidden7	32,768	SP8	ℓ_2	linear	25.01%	29.70%
hidden7	4,096	none	ℓ_2	RBF	27.53%	33.72%
hidden7	32,768	SP8	ℓ_2	RBF	29.41%	**34.95%**

Table 1: Various configurations of video classification. Spatiotemporal pooling: max pooling. A and B: 8- and 19-layer CNNs from [15] and [30], resp. SP8: the $1\times1+3\times1+2\times2$ and 3×1 spatial pyramids. Dataset: TRECVID MED'14 100Ex.

Layer	Dim.	Norm	SVM	mAP
output	2,000	root	linear	23.62%
output	2,000	root	χ^2	29.26%
hidden6	8,192	ℓ_2	linear	26.69%
hidden6	8,192	ℓ_2	RBF	33.33%
hidden7	8,192	ℓ_2	linear	27.41%
hidden7	8,192	ℓ_2	RBF	34.51%

Table 2: Objectness-guided pooling; CNN B; MED'14 100Ex.

Dim.	SVM/mAP	
	linear	RBF
32,768	29.70%	34.95%
4096	29.69%	34.55%
2048	29.02%	34.00%
1024	28.34%	33.18%

Table 3: Applying PCA on the top-perfoming feature in Table 1 prior to SVM. Feature: hidden7-layer feature extracted from CNN architecture B with SP8 and ℓ_2 normalization.

hidden7-SP8, shown in Table 3. Applying PCA reduced the feature dimension from 32,768 to 4096, 2048 and 1024 prior to SVM, which resulted in slightly reduced mAPs as compared to classification of features without PCA. However, it still outperformed or matched the performance of the features without spatial pyramids (CNN-B-hidden7-none in Table 1). Thus the spatial pyramid is effective in capturing useful information; and the feature dimension can be reduced without loss of important information.

5.3 Fusion Performance

We evaluated fusion of CNN features and Fisher vectors (FVs). The mAPs (and features dimensions) obtained with D-SIFT+FV and IDT+FV using RBF SVM classifiers were 24.84% and 28.45% (98,304 and 101,376), respectively. Table 4 reports our various fusion experiments. As expected, the late fusion of the static (D-SIFT) and motion-based (IDT) FVs brings a substantial improvement over the results obtained by the motion-based only FV. The fusion of CNN features does not provide much gain. This can be explained by the similarity of the information captured by the hidden layers and the output layer. Similarly, fusion of the hidden layer with the static FV does not provide improvement, although the output layer can still benefit from the late fusion with the static FV feature. However, fusion between any of the CNN-based features and the motion-based FV brings a consistent gain over the single best performer. This indicates that appropriate integration of motion information into the CNN architecture leads to substantial improvements.

5.4 Comparison with the State-of-the-Art

Table 5 shows the comparison of the proposed approach with strong hand-engineered approaches and other CNN-based approach on TRECVID MED'14. The proposed approach based on CNN features significantly outperformed strong hand-engineered approaches (D-SIFT+FV, IDT+FV and MIFS) even without integrating motion information. The CNN-

Features	mAP	vs. single feats	Features	mAP	vs. single feats
D-SIFT+FV, IDT+FV	33.09%	+8.25%, +4.64%	output, D-SIFT+FV	31.45%	+0.21%, +6.61%
			hidden6, D-SIFT+FV	31.71%	-1.83%, +6.87%
			hidden7, D-SIFT+FV	33.21%	-1.74%, +8.37%
output, hidden6	35.04%	+3.80%, +1.50%	output, IDT+FV	37.97%	+6.73%, +9.52%
output, hidden7	34.92%	+3.68%, -0.03%	hidden6, IDT+FV	38.30%	+4.76%, +9.85%
hidden6, hidden7	34.85%	+1.31%, -0.10%	hidden7, IDT+FV	**38.74%**	+3.79%, +10.29%

Table 4: Fusion. Used the best performing CNN features with kernel SVMs in Table 1.

Method	CNN	mAP
D-SIFT [22]+FV [26]	no	24.84%
IDT [34]+FV [26]	no	28.45%
D-SIFT+FV, IDT+FV (fusion)	no	33.09%
MIFS [16]	no	29.0 %
CNN-LCD$_{\text{VLAD}}$ with multi-layer fusion [39]	yes	36.8 %
proposed: CNN-hidden6	yes	33.54%
proposed: CNN-hidden7	yes	34.95%
proposed: CNN-hidden7, IDT+FV	yes	**38.74%**

Table 5: Comparison with other approaches on TRECVID MED'14 100Ex

based features have a lower dimensionality than the Fisher vectors. In particular, the dimension of the output layer is an order of magnitude smaller. The CNN architecture has been trained on high resolution images, whereas we applied it on low-resolution video frames which suffer from compression artifacts and motion blur. This further confirms that the CNN features are very robust, despite the domain mismatch. The proposed approach also outperforms the CNN-based approach by Xu *et al.* [39]. Developed independently, they used the same CNN architecture as ours, but different CNN layers, pooling, feature encoding and fusion strategies. The proposed approach outperforms all these competitive approaches and yields the new state-of-the-art, thanks to the carefully designed system based on CNNs and the fusion with motion-based features.

6 Evaluation in Action Recognition

6.1 Single Feature Performance

We evaluated our approach on a well-established action recognition dataset UCF-101 [31]. We followed the three splits in the experiments and report overall accuracy in each split and the mean accuracy across three splits. The configuration was the same as the one used in TRECVID MED'14 experiments, except that only linear SVM was used for fare comparison with other approaches on this dataset. The results are given in Table 6. The CNN architectures from [15] and [30] are referred as A and B. It shows that using hidden layer features yields better recognition accuracy compared to using softmax activations at the output layer. The performance also improves by up to 8% using deeper CNN architecture.

6.2 Fusion Performance

We extracted IDT+FV features on this dataset and obtained mean accuracy of 86.5% across three splits. Unlike the event detection task in TRECVID MED'14 dataset, the action recognition in UCF-101 is temporally trimmed and more centered on motion. Thus, the motion-based IDT+FV approach outperformed the image-based CNN-approach. However, as shown

CNN archi.	Features	Single feature accuracy split 1	split 2	split 3	mean	Fusion accuracy split 1	split 2	split 3	mean
A	output	69.02%	68.21%	68.45%	68.56%	85.57%	87.01%	87.18%	86.59%
A	hidden6	70.76%	71.21%	72.05%	71.34%	86.39%	87.36%	87.61%	87.12%
A	hidden7	72.35%	71.64%	73.19%	72.39%	86.15%	87.98%	87.74%	87.29%
B	output	75.65%	75.74%	76.00%	75.80%	87.95%	88.43%	89.37%	88.58%
B	hidden6	79.88%	79.14%	79.00%	79.34%	88.63%	90.01%	90.21%	**89.62%**
B	hidden7	79.01%	79.30%	78.73%	79.01%	88.50%	89.29%	90.10%	89.30%

Table 6: Performance of the proposed approach on UCF-101. Configurations are the same as in experiments on MED'14. Features are extracted with SP8, classified with linear SVMs.

Method	Mean accuracy over three splits
Spatial stream ConvNet [29]	73.0%
Temporal stream ConvNet [29]	83.7%
Two-stream ConvNet fusion by avg [29]	86.9%
Two-stream ConvNet fusion by SVM [29]	88.0%
Slow-fusion spatiotemporal ConvNet [14]	65.4%
Single-frame model [23]	73.3%
LSTM (image + optical flow) [23]	88.6%
MIFS [16]	89.1%
proposed: CNN-hidden6 only	79.34%
proposed: CNN-hidden6, IDT+FV (avg. fusion)	**89.62%**
proposed: CNN-hidden7, IDT+FV (avg. fusion)	89.30%

Table 7: Comparison with other approaches on UCF-101

in Table 6, a simple weighted average fusion of CNN-hidden6 and IDT+FT features boosts the performance to the best accuracy of 89.62%.

6.3 Comparison with the State-of-the-Art

Table 7 shows comparisons with other approaches. Note that the proposed image-based CNN-approach yields superior performance (6.3%, 6.0% and 13.9% higher) than the spatial stream ConvNet in [29], the single-frame model in [23] and the slow-fusion spatiotemporal ConvNet in [14] even though our model is not fine-tuned on specific dataset. The fusion performance of CNN-hidden6 (or CNN-hidden7) and IDT+FV features also outperforms the two stream CNN approach [29] and long short term memory (LSTM) approach that utilizing both image and optical flow [23]. The proposed fusion approach also outperforms the MIFS approach [16], which has shown to be competitive on the UCF-101 dataset. Note that both our image-trained and fusion approaches outperform the MIFS approach by a large margin on the temporally untrimmed and unconstraint TRECVID-MED 14 dataset.

7 Computational Cost

We have benchmarked the extraction time of the Fisher vectors and CNN features on a CPU machine. Extracting D-SIFT (resp. IDT) Fisher vector takes about 0.4 (resp. 5) times the video playback time, while the extraction of the CNN features requires 0.4 times the video playback time. The CNN features can thus be extracted in real time. On the classifier training side, it requires about 150s to train a kernel SVM event detector using the Fisher vectors, while it takes around 90s with the CNN features using the same training pipeline.

On the testing side, it requires around 30s to apply a Fisher vector trained event model on the 23,953 TRECVID MED'14 videos, while it takes about 15s to apply a CNN trained event model on the same set of videos.

8 Conclusion

In this paper we proposed a step-by-step procedure to fully exploit the potential of image-trained CNN architectures for video classification. While every step of our procedure has an impact on the final classification performance, we showed that CNN architecture, the choice of CNN layer, the spatiotemporal pooling, the normalization, and the choice of classifier are the most sensitive factors. Using the proposed procedure, we showed that an image-trained CNN architecture can outperform competitive *motion*- and *spatiotemporal*- based non-CNN approaches on the challenging TRECVID MED'14 video dataset. The result shows that improvements on the image-trained CNN architecture are also beneficial to video classification, despite the domain mismatch. Moreover, we demonstrated that adding some motion-information via late fusion brings substantial gains, outperforming other vision-based approaches on this MED'14 dataset. Finally, the proposed approach is compared with other approaches on the action recognition dataset UCF-101. The image-based approach outperforms other image-trained CNN approaches and the late fusion of image-trained CNN features and motion-based IDT-FV features is comparable with the state-of-the-art.

In this work we used an image-trained CNN as a black-box feature extractor. Therefore, we expect any improvements in the CNN to directly lead to improvements in video classification as well. The CNN was trained on the ImageNet dataset which mostly contains high resolution photographic images whereas the video dataset is fairly heterogeneous in terms of quality, resolution, compression artifacts and camera motion. Due to this domain mismatch, we believe that additional gains can be achieved by fine-tuning the CNN for the dataset. Even more improvements can possibly be made by learning motion information through a spatiotemporal deep neural network architecture.

9 Acknowledgements

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior National Business Center contract number D11PC20071. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC or the U.S. Government.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2911–2918, June 2012.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3286–3293. IEEE, 2014.

[4] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2, 2004.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision (ECCV)*, 2006.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European Conference on Computer Vision (ECCV)*, pages 392–407. Springer, 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision (ECCV)*, 2014.

[11] Mihir Jain, Hervé Jégou, and Patrick Bouthemy. Better exploiting motion for better action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2555–2562. IEEE, 2013.

[12] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[13] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo. Trajectory-based modeling of human actions with motion reference points. In *Computer Vision–ECCV 2012*, pages 425–438. Springer, 2012.

[14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Zhen-Zhong Lan, Ming Lin, Xuanchong Li, Alexander G. Hauptmann, and Bhiksha Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[17] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[18] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178, 2006. doi: 10.1109/CVPR.2006.68.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Zhe Lin, Zhuolin Jiang, and Larry S Davis. Recognizing actions by shape-motion prototype trees. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 444–451. IEEE, 2009.

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[23] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[25] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quéenot. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[26] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156, 2010.

[27] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[28] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[31] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, 2012.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. IEEE, 2015.

[34] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558. IEEE, 2013.

[35] Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference (BMVC)*, 2009.

[36] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2011.

[37] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[38] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2008.

[39] Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. A discriminative CNN video representation for event detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.