The Best of Both Worlds: Combining Data-independent and Data-driven Approaches for Action Recognition

Zhenzhong Lan, Dezhong Yao, Ming Lin, Shoou-I Yu, Alexander Hauptmann
{lanzzhzh, minglin, iyu, alex@cs.cmu.edu}, dyao@hust.edu.cn

Abstract

Motivated by the success of data-driven convolutional neural networks (CNNs) in object recognition on static images, researchers are working hard towards developing CNN equivalents for learning video features. However, learning video features globally has proven to be quite a challenge due to its high dimensionality, the lack of labelled data and the difficulty in processing large-scale video data. Therefore, we propose to leverage effective techniques from both data-driven and data-independent approaches to improve action recognition system.

Our contribution is three-fold. First, we propose a two-stream Stacked Convolutional Independent Subspace Analysis (ConvISA) architecture to show that unsupervised learning methods can significantly boost the performance of traditional local features extracted from data-independent models. Second, we demonstrate that by learning on video volumes detected by Improved Dense Trajectory (IDT), we can seamlessly combine our novel local descriptors with hand-crafted descriptors. Thus we can utilize available feature enhancing techniques developed for hand-crafted descriptors. Finally, similar to multi-class classification framework in CNNs, we propose a training-free re-ranking technique that exploits the relationship among action classes to improve the overall performance. Our experimental results on four benchmark action recognition datasets show significantly improved performance.

1. Introduction

Despite a long history of prior work, action recognition in videos, especially unconstrained videos that have large visual and motion variation, remains a very challenging task. Recent progress on this problem mainly relies on improvements of features, which can be categorized into two broad categories: 1) more traditional hand-crafted local features [38, 35] and their corresponding bag-of-feature (BoF) encoding methods [25], and 2) learning based features that are mainly inspired by the success of convolutional neural networks (CNNs) for image recognition [16, 31, 15] and recurrent neural networks (RNNs) for speech recognition [7, 8, 22]. In this paper we combine the merits of both classes of methodologies.

Trajectory based features, especially Improved Dense Trajectories (IDT) [37], are state-of-the-art hand-crafted features that have dominated action recognition on videos in recent years. Compared with other hand-crafted motion features, IDT performs better in that it models long term motion information and has a motion boundary descriptor (MBH) that is robust to camera motion. This long-term motion information modeling, as shown in [15, 31], is very hard to learn in a CNN framework. Despite its superiority, IDT still relies on simple hand-crafted local descriptors such as Histogram of Gradient (HOG) and Histogram of Optical Flow (HOF) [20] that took years of effort to develop. In contrast, for image and speech recognition
data-driven approaches have demonstrated their superiority and gradually replaced the traditional hand-crafted methods.

These revolutionary changes are largely enabled by the availability of neural networks algorithms, large scale labelled data and powerful parallel machines. Learning video features for action recognition, however, has proven to be quite a challenge due to its intrinsically high dimensionality, the lack of training data and the difficulty in processing large-scale video data [15, 31, 22]. With limited training data and computational power, the learned features are generally not discriminative enough and perform worse than IDT, especially on those datasets that have few training instances. Recent approaches [31, 22] circumvent these problems by learning on sampled frames or very short video clips and/or using weakly labeled data. However, video level label information can be incomplete or even missing at the frame/clip level and leads to the problem of false label assignment, which can be even worse for weakly labelled data [15]. In other words, the imprecise frame/clip-level labels populated from video labels are too noisy for learning powerful models. With better labelled data, neural network algorithms would give superior results. Unfortunately, accurately labeling video data is very expensive.

Though we see the value in developing fully automatically learned global video features using limited training data, we propose in this paper to revisit the traditional local feature pipeline and combine the merits of both data-independent and data-driven approaches. Inspired by the two-stream ConvNet [31], we introduce a two-stream Stacked Convolutional Independent Subspace Analysis (ConvISA) architecture to learn both visual appearance and motion information in an unsupervised way. As shown in Figure 1, instead of learning on frames/short video clips, we learn on much smaller primitives—video volumes that follow the trajectories detected by IDT. The learned descriptors, called LOP (Learned descriptors of Pixel) and LOF (Learned descriptors of Optical Flow), aim to improve the boost the performance of hand-crafted descriptors.

By following the traditional local feature pipeline, we can easily utilize techniques developed for traditional local descriptors to improve our data-driven descriptors.

Another merit CNN approaches have is that they naturally capture the relationships among action classes, which are hard to be captured by traditional ‘one-versus-all’ support vector machine (SVM) framework. To address this problem, we design a Multi-class Iterative Re-ranking (MIR) method. MIR requires no training yet can significantly improve the ranking performance measured by mean average precision (MAP). To make this technique useful for improving classification accuracy, we also propose a simple ranking-score fusion technique.

Our experimental results on four benchmark action recognition datasets show very competitive performance to the state-of-the-art results. Specifically, our methods achieve or exceed the state-of-the-art on HMDB51, Hollywood2, UCF101 and UCF50 datasets.

In the remainder of this paper, we provide more background information about video features with an emphasis on recent attempts on learning with deep neural networks. We then describe IDT and stacked convolutional ISA in detail. After that, we evaluate our methods. Further discussions including potential improvements are given at the end.

2. Related Work

In conventional video representations, features and encoding methods are the two chief reasons for considerable progress in the field. Among them, the trajectory based approaches [21, 34, 35, 37, 14], especially the Dense Trajectory (DT) method proposed by Wang et al. [35, 37], together with the Fisher Vector encoding [26] are the basis of the current state-of-the-art algorithms. Peng et al. [24] further improved the performance of DT by increasing the codebook sizes and fusing multiple coding methods. Sapiezyn et al. [30] explored ways to sub-sample and generate vocabularies for Dense Trajectory features. Hoai & Zisserman [10] achieved the state-of-the-art on several action recognition datasets by using three useful techniques including data augmentation, learning classifiers to model score distribution over video subsequences and learning classifiers to capture the relationship among action classes. Lan et al. [18] proposed a multi-skip feature stacking method that combines features at multiple frame rates to achieve temporal invariance for action features. They also proposed to attach normalized 3D location information to the local descriptors and achieve good results on several action benchmark datasets. Fernando et al. [5] proposed to model the evolution of appearance within the video and achieved state-of-the-art results on the Hollywood2 dataset.

Independent Component Analysis [12] (ICA) was the
first approach to learn representations of videos in an unsupervised way. Le et al. [19] approached this problem using stacked ConvISA, but they only model the appearance information without using optical flow information, which has struggled to capture motion information [15, 31]. Generative models for understanding transformations between pairs of consecutive images are also well studied. Recently, Ranzato et al. [27] proposed a generative model for videos. The model uses a recurrent neural network to predict the next frame or interpolate between frames.

More recently, some success has been reported using deep CNNs for action recognition in videos. Karpathy et al. [15] trained a deep CNNs using 1 million weakly labeled YouTube videos and reported a moderate success on using it as a feature extractor. Simonyan & Zisserman [31] reported a result that is competitive to IDT [37] by training deep CNNs using both sampled frames and stacked optical flows. Srivastava et al. [33] tried unsupervised feature learning using long-short term memory (LSTM) but get worse results than supervised feature learning methods.

As for modeling the relationship among classes, Bergamo & Torresani [1] proposed a Meta-class method to identify related image classes based on misclassification errors on a validation set. Hou et al. [11] proposed a method to identify similar class pairs and group them together to train ‘two versus rest’ classifiers. By combining ‘two versus rest’ with ‘one versus rest’ classifiers, they observed significant improvements from baselines. Hoai & Zisserman [10] proposed to learn the correlation and exclusion between action classes by learning to combine the base classifier’s prediction and sorted predictions from other classifiers. Unlike the aforementioned approaches that require learning and only modify the predictions once, our method is training free and iteratively updates the prediction scores given the better prediction scores in the previous iterations.

3. Improved Dense Trajectory

IDT improves DT feature [35] by explicitly estimating camera motion and removing trajectories generated by camera motion. In this section, we first describe how trajectories are generated and then the original hand-crafted descriptors.

3.1. Dense trajectories

As illustrated in Figure 1, trajectories are generated by tracking feature points across multiple frames. Gray pixel and optical flow volumes are extracted along the trajectories and represented by different kinds of descriptors. Following [35], our trajectories are extracted from multiple spatial-scale with a step size of $1/\sqrt{2}$ and a scale number of 8. Feature points are densely sampled with a step of size 5 pixels and tracked in each scale separately. Specifically, each feature point $f_t = (x_t, y_t)$ at frame $t$ is tracked to the $t+1$ frame by median filtering in a dense optical flow field. Mathematically,

$$f_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega)\hat{(x_t, y_t)}$$

(1)

where $M$ is the median filtering kernel, $(\hat{x}_t, \hat{y}_t)$ is the rounded position of $(x_t, y_t)$ and $\omega = (u_t, v_t)$ is the dense optical flow field calculated by the Farneback algorithm [4]. As can be seen, this tracking algorithm is computationally efficient given the optical flow field. The tracked points are put together to define a trajectory: $(f_t, f_{t+1}, f_{t+2}, \ldots f_{t+l-1})$, where $l$ is the length of the trajectory. Wang et al. [35] experimentally show that $l = 15$ gives good results across several benchmark datasets and longer trajectories may suffer from the drifting problem. Static trajectories are non-informative and hence removed.

3.2. Hand-crafted descriptors for IDT

There are four hand-crafted descriptors for IDT. Among them, trajectory shape, which encodes the relative locations of the trajectories, can directly get from $f_t$. The other three descriptors are Histogram of Gradient (HOG), Histogram of Optical flow (HoF) and MBH. IDT computes HOG/HOF along the dense trajectories. For both HOG and HOF, full orientations are quantized into 8 bins with an additional zero bin for HOF. Both descriptors are normalized with their $\ell_2$-norm. The MBH descriptor separates the optical flow field into its $x$ and $y$ components. Spatial derivatives are computed for each of them and orientation information is quantized into histograms, similar to the HOG descriptor. For each component, there is a 8-bin histogram normalized by its $\ell_2$-norm. Since MBH represents the gradient of the optical flow, constant motion information is suppressed and only information about changes in the flow field is kept, which is a simple and effective way to remove camera motion. These histogram-based descriptors are computed within space-time volumes aligned with a trajectory to encode the appearance and motion information. The size of the volume is $s \times s$ pixels and $l$ frames long, which corresponds to the input size of stacked ConvISA. To embed structure information, the volume is subdivided into a spatio-temporal grid of size $s_{\tau} \times s_{\tau} \times l_{\pi}$. In this paper, we fix the size of volume and grid as in IDT, i.e., $s = 32$, $l = 15$, $s_{\tau} = 2$ and $l_{\pi} = 3$ as in [35].

4. Learning Descriptors for IDT

Although IDT is the current state-of-the-art action recognition feature, it still relies on hand-crafted descriptors. In this section, we will describe how we use Stacked ConvISA to learn descriptors for IDT. By learning descriptors for IDT, we also aim to generalize the best performing hand-crafted features within a data-driven learning framework.
4.1. Stacked ConvISA

ISA is an unsupervised learning algorithm that learns features from unlabeled image patches. An ISA network [13] can be described as a two-layered network. More precisely, let matrix $W \in \mathbb{R}^{m \times n}$ and matrix $V \in \mathbb{R}^{d \times m}$ denote the parameters of the first and second layers of ISA respectively. $n$ is the dimension of the input matrix and $d$ is the dimension of outputs of ISA. $m$ is the number of latent variables between the first layer and the second layer. Typically $d \leq m \leq n$. The matrix $W$ is learned from data with orthogonal constraint $WW^\top = I$. Therefore we call $W$ the projection matrix. The matrix $V$ is given by the network structure to group the output variables of the first layer. $V_{ij} = 1$ if the $j$-th output variable of the first layer is in the $i$-th group, otherwise $V_{ij} = 0$. Therefore we call $V$ the grouping matrix. Given an input pattern $X^t \in \mathbb{R}^n$, the activation of $i$-th output unit of the second layer is $p_i(X^t; W, V)$ defined by

$$p_i(X^t; W, V) \triangleq \sqrt{\sum_{k=1}^{m} V_{ik} \left( \sum_{j=1}^{n} W_{kj} X_j^t \right)^2}.$$  

(2)

The ISA enforces the activation of the output unit to be sparse. To achieve the sparse activation, it minimizes the loss function defined on $T$ training instances:

$$\min_W \sum_{t=1}^{T} \sum_{i=1}^{d} p_i(X^t; W, V_i) \quad \text{s.t.} \quad WW^\top = I.$$  

(3)

Another way to interpret ISA is from sparse coding framework. Let $\mathcal{G} = [\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_d]$ denote the variable group indexes defined by $V$, that is, $j \in \mathcal{G}_i$ if and only if $V_{ij} = 1$. $|\mathcal{G}_i|$ defines group size, which is generally the same across groups.

As in group LASSO [6], for any vector $a \in \mathbb{R}^m$, we defined the group $\ell_1$-norm $\|a\|_{\mathcal{G}, 1}$ as

$$\|a\|_{\mathcal{G}, 1} \triangleq \sum_{i=1}^{d} \sqrt{\sum_{j \in \mathcal{G}_i} a_j^2}.$$  

We can write $p_i(X^t; W, V)$ as

$$p_i(X^t; W, V) = \| WX^t \|_{\mathcal{G}, 1}.$$  

Denote $\alpha_t = WX^t$, since $WW^\top = I$, we have

$$X^t = W^\top \alpha_t,$$

where $W^\top$ is the Moore–Penrose pseudo inverse of $W$. Eq. (3) can be re-formulated as a sparse coding method that

$$\min_{W, \alpha_t} \sum_{t=1}^{T} \|\alpha_t\|_{\mathcal{G}, 1} \quad \text{s.t.} \quad (W^\top)^\top W^\top = I \quad X^t = W^\top \alpha_t.$$  

(4)

Based on Eq. (4), ISA is essentially searching a group-sparse representation $\alpha_t$ of the input signal $X_t$. The matrix $W^\top$ is the dictionaries of sparse coding. The orthogonal constraint of $W^\top$ makes the learned components maximally independent.

On high dimensional data, ISA is computationally expensive to run. In ISA, we usually set $m = O(n)$. The optimization defined in Eq. (3) will be computationally expensive for high dimensional data since its space complexity is $O(n^2)$ and time complexity is $O(n^3)$. In order to handle this problem, Le et al. [19] proposed a stacking architecture that progressively makes use of PCA and ISA as sub-units for unsupervised learning. The key ideas of this approach are as follows. An ISA algorithm on small input patches was first trained. This learned network is then used to convolve with a larger region of the input image with a fix stride (step size). The combined responses of the convolution step are then given as input to the next layer which is also accomplished by another ISA algorithm with PCA as a preprocessing step. Similar to the first layer, PCA is used to whiten the data and reduce their dimensions such that the next layer of the ISA algorithm only works with low dimensional inputs. The stacked model is trained greedy layer-wise in the same manner as other algorithms proposed in the deep learning literature [9]. Applying the models above to the video domain is rather straightforward: the inputs to the network are 3D video blocks instead of image patches. More specifically, a sequence of image patches are flatten into a vector. This vector becomes input features to the network above. The detail algorithm of stacked ConvISA can be found in Algorithm 1. The parameters are set as in [19].

4.2. Two stream stacked ConvISA

The initial stacked ConvISA was designed to learn appearance and motion together, but as shown in [15], spatio-temporal features learned from fixed size cuboids do not capture the motion part well [31]. Inspired by two stream ConvNet[31], we propose a two-stream stacked ConvISA that learn appearance features and motion features separately. The appearance stream learns on gray scale video volumes that follow the trajectories detected by IDT and the motion stream learns to describe the corresponding optical flow volumes. In this paper, we use a two-layer stacked ConvISA as in [19]. The inputs of the stacked ConvISA are $32 \times 32 \times 15$ video volumes, within which we subsample $16 \times 16 \times 5$ video volumes to train the first layer of stacked ConvISA. Figure 2 shows some randomly chosen filters learned from two stream stacked ConvISA. As can be seen, filters from optical flow have much more complex shapes than filters from gray pixels, which indicates that motion signals are more difficult to learn.
Algorithm 1 Stacked ConvISA

1: **Input:** Sample T trajectory volumes \( X = [X^1, X^2, \ldots, X^T] \) from videos, where \( X^i \in \mathbb{R}^{32 \times 32 \times 15} \).
2: —— First layer network ——
3: Uniformly sample T sub-volumes \( \hat{X} = [\hat{X}^1, \hat{X}^2, \ldots, \hat{X}^T] \) from \( X \), where \( \hat{X}^i \in \mathbb{R}^{16 \times 16 \times 5} \).
4: Train PCA on \( \hat{X} \) with PCA dimension of 300:
   \[ \text{pca}_1 = \text{PCA}_{\text{train}}(\{\hat{X}^1, \ldots, \hat{X}^T\}, 300) \]
   \[ \hat{X}_{\text{pca}_2} = \text{PCA}_{\text{apply}}(\text{pca}_1, \hat{X}) \]
5: Train ISA on \( \hat{X}_{\text{pca}_2} \) with ISA group size 1:
   \[ \text{isa}_1 = \text{ISA}_{\text{train}}(\hat{X}_{\text{pca}_2}, 1) \]
6: Convolve \( X \) with first layer’s model:
   \[ X_{\text{ISA}_1} = \text{ISA}_{\text{apply}}(\text{isa}_1, X) \]
7: Using \( X_{\text{ISA}_1} \) as the output of the first layer network:
   \[ X_{\text{layer1}} = [X_{\text{ISA}_1}] \]
8: —— Second layer network ——
9: Train PCA on \( X_{\text{layer1}} \) with PCA dimension of 200:
   \[ \text{pca}_2 = \text{PCA}_{\text{train}}(X_{\text{layer1}}, 200) \]
   \[ X_{\text{pca}_2} = \text{PCA}_{\text{apply}}(\text{pca}_2, 2, X_{\text{layer1}}) \]
10: Train ISA on \( X_{\text{pca}_2} \) with group size 2:
   \[ \text{isa}_2 = \text{ISA}_{\text{train}}(X_{\text{pca}_2}, 2) \]
   \[ X_{\text{ISA}_2} = \text{ISA}_{\text{apply}}(\text{isa}_2, X_{\text{pca}_2}) \]
11: Stack the top 100 dimension of \( X_{\text{pca}_2} \) with \( X_{\text{ISA}_2} \) as the output \( X_{\text{layer2}} \) of the second layer network:
   \[ X_{\text{layer2}} = [X_{\text{pca}_2}(1:100); X_{\text{ISA}_2}] \]
12: **Output:** \( X_{\text{layer2}} \)

5. Multi-class Iterative Re-ranking (MIR)

Algorithm 2 shows our MIR algorithm. Given a score matrix \( P \in \mathbb{R}^{N \times K} \) that contains K classifiers’ predictions on N instances. We update each score \( P_{i,j} \) iteratively by looking at other classifiers’ predictions on the same instance and reducing the score using the predictions from the other classifiers. The reduction is carried out by first sorting other classifiers’ predictions \( \{P_{i,1}^{(w)}, P_{i,2}^{(w)}, \ldots, P_{i,K}^{(w)}\} \) of \( P_{i,j}^{(w)} \) in a descending order and then subtracting the weighted sum of the sorted scores from \( P_{i,j} \). We use exponential decaying weight and the weighting coefficient \( \alpha > 0 \) has been set to 1 throughout the paper. MIR assumes that most instances only belong to one class. We acknowledge that this assumption is quite strong in most real-world applications. However, our experiments show that even with this assumption, MIR is quite useful for improving ranking performance for real-world data where each instance can have multiple labels. We found algorithm 2 only improve the ranking performance but not classification accuracy. To make this improvement on ranking be beneficial for classification accuracy, we propose to fuse the improved ranking from MIR with the original scores. Specifically, after getting the improved ranking matrix \( P_{R} \) from \( P_{(W)} \), we normalize \( P_{R} \) by scaling between 0 and 1 and do a z-score normalization to get \( P_{R} \), which becomes comparable to the original score matrix \( P \). Our final prediction would be \( P = P_{R} + P \).

6. Experiments

We examine our the proposed LOP and LOF descriptors on several activity recognition datasets, predominately in-

![Algorithm 1 Stacked ConvISA](image1)

![Algorithm 2 Multi-class Iterative Re-ranking](image2)

![Example filters learned using two-stream stacked ConvISA.](image3)
volving actions. The experimental results show that the new descriptors, together with several existing feature enhancing techniques and MIR, can improve IDT representations significantly.

6.1. Experimental setting

As in [37] and [19], IDT features are extracted using 15 frame tracking, camera motion stabilization and RootSIFT normalization and described by Trajectory, HOG, HOF, MBH, LOP and LOF descriptors. Stacked ISA models are trained on 200000 video volumes. PCA is used to reduce the dimensionality of descriptors by a factor of two. For Fisher Vector encoding, we map the raw descriptors into a Gaussian Mixture Model with 256 Gaussians trained from a set of randomly sampled 256000 data points. Power and $\ell_2$ normalization are also used before concatenating different types of descriptors into a video based representation. For classification, we use a linear SVM classifier with a fixed $C = 100$ as recommended by [37] and the one-versus-all approach is used for multi-class classification scenario. We use two feature enhancing techniques developed by Lan et al. [18] to show that improvements developed for hand-crafted descriptors can also be used for our learned descriptors. These two techniques are is 3D location extension (xyt-extension) and Multi-skip Feature Stacking (MIFS). MIR is used after xyt-extension and MIFS are applied.

6.2. Datasets

Four representative datasets are used: The HMDB51 dataset [17] has 51 action classes and 6766 video clips extracted from digitized movies and YouTube. [17] provides both original videos and stabilized ones. We only use original videos in this paper. The Hollywood2 dataset [20] contains 12 action classes and 1707 video clips that are collected from 69 different Hollywood movies. We use the standard splits with training and testing videos provided by [20]. The UCF101 dataset [32] has 101 action classes spanning over 13320 YouTube videos clips. We use the standard splits with training and testing videos provided by [32]. The UCF50 dataset [29] has 50 action classes spanning over 6618 YouTube videos clips that can be split into 25 groups. The video clips in the same group are generally very similar in background. Leave-one-group-out cross-validation as recommended by [29] is used. We report mean accuracy (MAcc) for HMDB51, UCF101 and UCF50 and mean average precision (MAP) for Hollywood2 datasets as in the original papers.

6.3. Evaluation of our learned IDT descriptors

In this section we compare hand-crafted descriptors with our learned descriptors as well as their combination. To compute the descriptors, we follow the parameter settings as in [35] and [19]. That is, $s = 32, s_r = 2, l_r = 3$ for all descriptors. We fix the trajectory length to $l = 15$. We use $d_1 = 300$ as the dimension of first layer ISA outputs and $d_2 = 200$ as the dimension of the second layer ISA outputs. In section 6.7, we will show that these parameters experimentally give good performance cross datasets.

Results of the four datasets are presented in Table 1. Overall, our learned descriptors can boost the performance of IDT by 1% to 3%. Comparing LOP to HOG, we can see that LOP is doing much better than HOG in capturing appearance information, resulting in 3% to 8% improvement. Our LOF descriptor is better than HOF, but is still worse than MBH. These results show that even learning on video volumes defined by trajectories, motion information is hard to capture and represent. When comparing the combination of hand-crafted features and the combination of learned features, our learned features are generally worse than hand-crafted features except UCF101 datasets, but when combining all descriptors, our learned descriptors can boost the performance significantly. These improvements demonstrate that unsupervised learning methods have the potential to significantly boost the performance of hand-crafted features.

6.4. Feature enhancing techniques

We apply two feature enhancing techniques by Lan et al. [18] to improve the performance of our hybrid descriptors that including hand-crafted descriptors and learned descriptors. The first technique is 3D location extension (xyt-extension), which augments the descriptors with 3D normalized location information. Another technique is Multi-skip Feature Stacking (MIFS), which stack raw features extracted from videos with different frame rates before encoding. Table 2 summarized the results. Overall, both xyt-
extension and MIFS help to improve the performance of our hybrid descriptors on all four datasets. These improvements show that by learning on local video volumes defined by IDT, we can easily utilize available techniques developed for hand-crafted features, which would be difficult or not possible for global feature learning to do.

### 6.5. Multi-class iterative re-ranking (MIR)

In this section, we report results of our MIR technique on the combined results in section 6.4. Table 3 shows the number of iteration versus MAP performance on four datasets we tested. As can be seen, on all four datasets, MIR finishes within 4 iterations and converges to results that are significantly better than the baselines, which are shown in iteration 0. These improvements show that although MIR may be based on strong assumptions, it is still quite useful for real-world data. These improvements are especially seen with Hollywood2 dataset, whose source is movie data where multiple labels exist for some instances. For HMDB51, UCF101 and UCF50, after fusing the score with ranking, we get slight improvements on MAcc, from 66.5, 89.7 and 95.1 to 67.0, 90.2 and 95.4, respectively.

### 6.6. Comparison to the state-of-the-art

In Table 4, we compare our best results with the state-of-the-art approaches. From Table 4, in most of the datasets, we observe improvements over the state-of-the-art. Note that although we list several most recent approaches here for comparison purposes, most of them are not directly comparable to ours due to the use of different data augmentation, features and representation methods. For example, Hoai & Zisserman [10] and Fernando et al. [5] use left-right mirrored video data augmentation to get about 2% improvements on Hollywood2 dataset, which can be easily adopted by our algorithms while they can also use the MIFS and xyt extension techniques we used in this paper. The most comparable one is Lan et al. [18], with which we enhance our approaches. Karpathy [15] trained ConvNets on 1 million weakly labeled YouTube videos and reported 65.4% MAcc on UCF101 datasets. Simonyan & Zisserman [31] reported results that are competitive to IDT by training deep convolutional neural networks using both sampled frames and optical flows and get 59.4% MAcc in HMDB51 and 88.0% MAcc in UCF101, which are comparable to the results of Wang & Schmid [37]. Peng et al. [25] achieves similar results to ours on the HMDB51 datasets by combining a hierarchical Fisher Vector with the original one.

### 6.7. Evaluation of parameters

We now move on to our characterization of performance on various axes of parameters for our new descriptors. We report results on HMDB51 and Hollywood2, as they are the most of them are not directly compa-
two most challenging datasets according to performance. We study the effect of stride, receptive field size (size of first layer input) and number of features (output size) as in [3]. We evaluate the performance for a parameter at a time. The other parameters are fixed as default values.

**Effect of stride** First, we evaluate the effect of stride. In this experiment, we test spatial strides of 4, 8, and 16 pixels, and temporal strides of 2 and 5. The results are summarized in Figure 3. Surprisingly, we find that, different from [3], larger strides consistently give better performance. Since we observe a constant performance improvement by increasing the spatial stride from 4 to 16 and the temporal stride from 2 to 5, we attempted to increase to an even larger stride. However, the stride size is constrained by the size of the input volume. Results in Figure 4 confirms that the reason for larger strides giving better results is that they cause less overlap in convolution and hence provide less redundant information to the next layer network. It is a notable fact since non-overlap scanning saves large amounts of training and prediction time.

**Effect of receptive field** We also study the effect of receptive field size, i.e., the input size of the first layer of the network. We test spatial sizes of 8, 16 and 24, and temporal sizes of 5 and 10 and use the maximum stride that can cover the input volumes. For example, for a respective field of $8 \times 8 \times 10$, the stride would be $(8, 5)$. As can be seen in Figure 4, the receptive field size does not affect the performance as much as the stride. Overall, the $(16, 5)$ pair works best. Meanwhile, larger receptive fields do not give better performance. Thus, we suggest using $(16, 5)$ as it is a good balance between computational cost and performance.

**Dimension of features** Finally, we study the dimension of features ranging from 50 to 300 with a step size of 50. The dimension of features balances the information that is lost and the noise that has been reduced. As can be seen in Figure 5, the dimension of 200 is a good trade-off given the size of inputs we have. This result is consistent with what has been suggested by [19].

### Table 4: Comparison of our results to the state-of-the-art.

| Method             | HMDB51 (MAcc. %) | Hollywood2 (MAP %) | UCF101(MAcc. %) | UCF50 (MAcc. %) |
|--------------------|------------------|---------------------|-----------------|-----------------|
| Wang et al. [37]   | 57.2             | 63.3                | 65.4            | 90.0            |
| Simonyan et al. [31]| 59.4             | 64.3                | 85.9            | 90.0            |
| Peng et al. [24]   | 61.1             | 68.0                | 87.9            | 91.2            |
| Lan et al. [18]    | 65.0             | Hoai et al. [10]    | 88.0            | 92.3            |
| Peng et al. [25]   | 66.8             | Fernando et al. [5] | 89.1            | 94.4            |
| Ours               | 67.0             | Ours                | 90.2            | Ours            |

### Table 5: Generalizability of Stacked ConvISA models.

| Train/Apply | HMDB51 | Hollywood2 |
|-------------|--------|------------|
| HMDB51      | 56.1   | 59.3       |
| Hollywood2  | 56.0   | 61.1       |

### 6.8. Model generalizability

CNN models have been shown to generalize well across different datasets [28]. It would also be interesting to see the generalizability of our unsupervised learned models. To show this, we apply the model learned from Hollywood2 dataset to HMDB51 dataset and vice versa. The results are shown in Table 5, from which we can see that if we train on Hollywood2 and apply it to HMDB51, there is almost no performance lost while if we train on HMDB51 and apply the model to Hollywood2, the performance drops by about 2%. These results indicate that if we train on difficult datasets like Hollywood2, the model can generalize well.

### 7. Conclusions

Different from current trend of learning video feature using deep neural networks, which is computationally intensive and label demanding, we propose in this paper to revisit the traditional local feature pipeline and combine the merits of both data-independent and data-driven approaches. As an example, we present two novel local video descriptors for IDT, a state-of-the-art local video feature. The proposed descriptors are learned using Stacked ConvISA on gray pixel and optical flow volumes that follow the trajectories detected by IDT. We also design a multi-class iterative re-ranking technique called MIR to exploit relationships among classes. Extensive experiments on four real-world datasets show that our new descriptors, when combined with hand-crafted descriptors and using MIR, achieve or exceed state-of-the-art methods. Future work would be determining the appropriate depth for Stacked ConvISA. Additionally, we would like to study other unsupervised learning methods for video feature learning.
References

[1] A. Bergamo and L. Torresani. Meta-class features for large-scale object categorization on a budget. In CVPR, 2012.

[2] A. Ciptadi, M. S. Goodwin, and J. M. Rehg. Movement pattern histogram for action recognition and retrieval. In ECCV, 2014.

[3] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In AISTATS, 2011.

[4] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In Image Analysis, pages 363–370. Springer, 2003.

[5] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, T. Tuytelaars, and L. Belgum. Modeling video evolution for action recognition. In CVPR, 2015.

[6] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. arXiv preprint arXiv:1001.0736, 2010.

[7] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In ICML, 2014.

[8] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In ICASSP, 2013.

[9] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. In ICML, 2006.

[10] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In ACCV, 2014.

[11] R. Hou, A. R. Zamir, R. Sukthankar, and M. Shah. Damm–discriminative and mutually nearest: Exploiting pairwise category proximity for video action recognition. In ECCV, 2014.

[12] J. Hurri and A. Hyvärinen. Simple-cell-like receptive fields maximize temporal coherence in natural video. Neural Computation, 15(3):663–691, 2003.

[13] A. Hyvärinen, J. Hurri, and P. O. Hoyer. Natural Image Statistics: A Probabilistic Approach to Early Computational Vision., volume 39. Springer Science & Business Media, 2009.

[14] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In ECCV, 2012.

[15] 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.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

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

[18] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. CVPR, 2015.

[19] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In CVPR, 2011.

[20] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In CVPR, 2009.

[21] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In ICCV Workshops, 2009.

[22] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and M. Toderici. Beyond short snippets: Deep networks for video classification. arXiv preprint arXiv:1503.08909, 2015.

[23] D. Onetea, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In ICCV, 2013.

[24] X. Peng, L. Wang, X. Yang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. arXiv preprint arXiv:1405.4506, 2014.

[25] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In ECCV, 2014.

[26] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In ECCV, 2010.

[27] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604, 2014.

[28] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In CVPR Workshop, 2014.

[29] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. Machine Vision and Applications, 24(5):971–981, 2013.

[30] M. Sapienza, F. Cuzzolin, and P. H. Torr. Feature sampling and partitioning for visual vocabulary generation on large action classification datasets. arXiv preprint arXiv:1405.7545, 2014.

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

[32] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012.

[33] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. arXiv preprint arXiv:1502.04681, 2015.

[34] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In CVPR, 2009.

[35] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition through dense trajectories. In CVPR, 2009.

[36] H. Wang and C. Schmid. Lear-inria submission for the thumos workshop. In ICCV Workshop, 2013.

[37] H. Wang, C. Schmid, et al. Action recognition with improved trajectories. In ICCV, 2013.

[38] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In BMVC, 2009.