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

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

Abstract

Motivated by the success of CNNs in object recognition on images, researchers are striving to develop CNN equivalents for learning video features. However, learning video features globally has proven to be quite a challenge due to the difficulty of getting enough labels, processing large-scale video data, and representing motion information. 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 explicitly show that local handcrafted features and CNNs share the same convolution-pooling network structure. Second, we propose to use independent subspace analysis (ISA) to learn descriptors for state-of-the-art handcrafted features. Third, we enhance ISA with two new improvements, which make our learned descriptors significantly outperform the handcrafted ones. Experimental results on standard action recognition benchmarks show competitive performance.

1. Introduction

Despite a long history of prior work, action recognition in videos, especially unconstrained videos with large visual and motion variations, remains a challenging task. Recent progress on this problem mainly relies on improvements of features, which can be categorized into two broad classes: 1) more traditional hand-crafted local features [35, 32] and their corresponding bag-of-feature (BoF) encoding methods [24], and 2) learning based features that are mainly inspired by the success of convolutional neural networks (CNNs) for image recognition [15, 27, 14] and of recurrent neural networks (RNNs) for speech recognition [5, 6, 21]. In this paper we combine the merits of both methodologies.

Trajectory based features, especially Improved Dense Trajectories (IDT) [34], are state-of-the-art hand-crafted features that have dominated action recognition in videos over 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) which is robust to camera motion. This long-term motion information modeling, as shown in [14, 27], is very hard to learn in a CNN framework. Despite its superiority, IDT, somewhat surprisingly, 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 [15, 21], data-driven approaches have consistently demonstrated their superiority and have been gradually replacing the traditional hand-crafted methods.

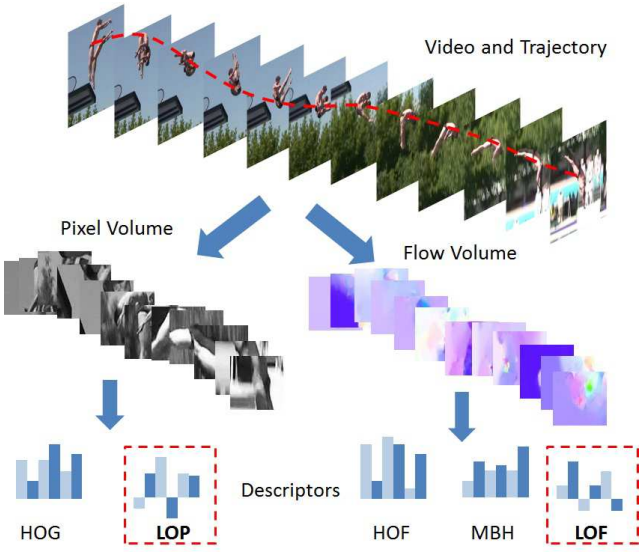


Figure 1: Illustration of our novel local video descriptors. LOP and LOF describe gray pixel and optical flow volumes, respectively. They resemble HOG/HOF/MBH in a data-driven learning framework.

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 intrinsic high dimensionality, the lack of training data, and the difficulty in processing large-scale video data [14, 27, 21]. With limited training data

and computational power, the learned features are generally not discriminative enough and perform worse than IDT, especially among datasets that have few training instances. Recent approaches [27, 21] circumvent these problems by learning from sampled frames or very short video clips, as well as using weakly labelled data. However, video-level label information can be incomplete or even missing at the frame/clip-level and leads to false label assignment, which can be even worse for weakly labelled data [14]. In other words, the imprecise frame/clip-level labels populated from video labels are usually too noisy for learning powerful models. With better labelled data, neural network algorithms can give superior results. Unfortunately, accurately labelled video data is very expensive to obtain.

Though we see the value in developing fully automatically learned global video features using labeled training data, in this paper we propose to revisit the traditional local feature pipeline and unsupervised feature learning methods, connect both data-independent and data-driven approaches, and combine their strengths. Inspired by the two-stream ConvNet [27] and ConvISA [19], we introduce a two-stream ISA-IDT to learn both visual appearance and motion information in an unsupervised manner. As shown in Figure 1, instead of learning from frames or short video clips, we learn from 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 best performing hand-crafted descriptors within an unsupervised data-driven learning framework. Our proposed architecture has several attractive properties:

- Compared to full video learning, small video volumes lie in a much lower dimensional space, hence they are computationally efficient to learn and apply.
- Unsupervised learning avoids the costly work of collecting labelled data and the false label assignment problem among current supervised video learning settings.
- Through learning from video volumes defined by IDT, the resulting descriptors can be seamlessly combined with hand-crafted descriptors and boost the overall performance.
- By following the traditional local feature pipeline, we can easily utilize techniques developed for traditional local descriptors to improve our data-driven descriptors.

Although the idea of unsupervised video feature learning sounds appealing, it is, in fact, a very challenging problem. It introduces several novel problems that have not been sufficiently studied in the literature. The first one, of course, is the challenge of achieving high accuracy. For our algorithm to be useful, it needs to at least as good as IDT. This is by no means easy. For example, in unsupervised image feature learning, after years of research efforts, SIFT was still the best ingredients in PASCAL VOC challenges 2012 ([2]). The second challenge, which is unique to video feature learning, is how to learn to describe optical flow data in an unsupervised way. Research in the past [18, 1, 27, 34] show that the optical flow feature is an essential part of motion representation. To the best of our knowledge, we are the first to deal with unsupervised optical flow feature learning.

Before revealing how we address the above mentioned challenges, let us first show that our algorithm indeed outperforms IDT. We conduct experiments on the benchmark action datasets of HMDB51 and UCF101, as in [27]. Table 1 compares the model training time and accuracy of our method to IDT, as well as the two-stream CNN ([27]), a state-of-the-art CNN approach. Note that for both IDT and two-steam CNN, several improvements have been proposed since they were first introduced in 2013 and 2014, respectively; but we compare results from the original papers as most of the improvements can also be applied to our method. Later in this paper, we will have more complete comparisons to the state of the art. As can be seen in Table 1, in terms of training time, our approach is much more efficient than two-stream CNNs by several orders of magnitude. Two-stream CNNs need about 1 day to train a model on 4 Titan-X GPUs while our method only needs about 2 hours on 1 CPU. IDT feature training only needs around 1 hour on 1 CPU because the only part that requires learning is the codebook training. With regard to accuracy, our method outperforms two-stream CNNs on HMDB51 and has similar results on UCF101 despite the fact that it was trained on less data and does not need any labels to train the feature extraction module. Our results are also significantly better than the results of IDT.

In the remainder of this paper, we first provide more background information about video features with an emphasis on recent attempts at learning with deep neural networks. We then describe the relationship between hand-crafted features and CNN-based features in detail, followed by the descriptions of our two-stream ISA-IDT algorithm. After that, we conduct experiments and show more detailed comparisons of our method to the state-of-the-art methods. Further discussions including potential improvements are provided at the end.

2. Related Work

Features and encoding methods are the major sources of breakthroughs in conventional video representations. Among them, trajectory based approaches [34, 12], especially the Dense Trajectory (DT) and IDT [32, 34], are

	Feature Training Time	Need Label	HMDB51	UCF101
Ours	∼2 hours / 1 CPU	No	**61.5%**	**88.3%**
IDT	∼ 1 hour / 1 CPU	No	57.2 %	85.9%
Two-stream CNNs	∼ 1 day / 4 GPUs	Yes	59.4%	88.0%

Table 1: Performance comparison of our approach with IDT and two-stream CNNs.

the basis of current state-of-the-art hand-crafted algorithms. These trajectory-based methods are designed to address the flaws of image-extended video features. Their superior performance validates the need for a unique representation of motion features.

There have been many studies attempting to improve IDT due to its popularity. Peng et al. [23] enhanced the performance of IDT by increasing codebook sizes and fusing multiple coding methods. Sapienza et al. [26] explored ways to sub-sample and generate vocabularies for DT features. Hoai & Zisserman [7] achieved state-of-the-art performance on several action recognition datasets by using three techniques including data augmentation, modeling score distribution over video subsequences, and capturing the relationship among action classes. Fernando et al. [3] modeled the evolution of appearance in the video and achieved state-of-the-art results on the Hollywood2 dataset. [17] proposed to extract features from videos with multiple playback speeds to achieve speed invariances. However, none of them dealt with the fact that IDT relies on very simple, hand-crafted descriptors. In contrast, many data-driven approaches have demonstrated their modeling power in image recognition [15] and are graduallyquickly replacing traditional hand-crafted methods.

Motivated by this success of CNNs, researchers are working intensely towards developing CNN equivalents for learning video features. Several accomplishments have been reported from using CNNs for action recognition in videos [38, 37, 31]. Karpathy et al. [14] trained deep CNNs through one million weakly labelled YouTube videos and reported moderate success while using it as a feature extractor. Simonyan & Zisserman [27] demonstrated a result competitive to IDT [34] through training deep CNNs using both sampled frames and stacked optical flows. Wang et al. [36] use the outputs of two-stream CNNs as features to replace HOG and achieve state-of-the-art results on HMDB51 and UCF101 datasets. All of the above relied on a large amount of labels which are expensive to get and generally perform worse than hand-crafted features among small datasets.

There have been a limited number of studies regarding unsupervised methods for learning video features. Among them the Independent Component Analysis (ICA) [8] was the first approach to learn representations of videos in an unsupervised way. Le et al. [19] addressed the issue using stacked ConvISA. Srivastava et al. [29] applied unsupervised feature learning through long-short term memory. Since these methods rely purely on pixel data, they struggled to capture motion information and generally performed no better than state-of-the-art hand-crafted methods. Also, the network structures of these methods, because they are designed for pixel data, cannot directly used in learning motion features.

There are also several attempts at connecting the traditional feature encoding pipeline to the neural network frameworks. Vladyslav et al. [30] studied the structure similarities between Fisher vectors and neural networks and proposed to jointly optimize Fisher vectors and the classifier. Richard and Gall [25] converted the kMeans-based BoW model into an equivalent recurrent neural network and trained the BoW model and classifier together. Both above approaches focus on the end-to-end training of CNNs and again require labels and significantly increase the model training time. Instead, we emphasize the convoltional-pooling structure of CNNs rather than their training methods. Jarrett et al. [11] also pointed out the connection between handcrafted features and one stage CNNs. However, they focus on image feature learning, which is inherently different from video feature learning. They also did not explicitly explain what linear and non-linear operators these handcrafted features have and how to map them into a CNN framework.

This study overcomes many limitations from previous works by designing and adapting unsupervised feature learning methods to video and optical flow volumes detected by IDT. Our new learning paradigm does not rely on any label, hence can work well among small datasets. It is better at capturing motion information due to our enhanced approaches to model optical flow information, and can use feature enhancing techniques developed for hand-crafted descriptors, as illustrated by MIFS.

3. Improved Dense Trajectory

IDT improves DT feature [32] through explicitly estimating camera motions and removing trajectories generated by them. It relies on histogram-based descriptors, which 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 ISA. To embed structure information, the volume is subdivided into a spatio-temporal grid of size $s_\tau \times s_\tau \times l_\pi$. The default size

of volume and grid for IDT are $s = 32$, $l = 15$, $s_\tau = 2$ and $l_\pi = 3$.

4. The Convolution-Pooling Architecture

In this section we first define the convolution-pooling structure and then compare IDT with CNN-based video features. We highlight their structural similarities by showing that they are both features generated by deep convolution-pooling cascade with two key elements: convolution and pooling layers.

We define a convolution-pooling cascade as any single, iterative or recursive implementation of the following sequence of operations:

$$c(x) = f(w \otimes x)$$

$$p(x) = g(c(x))$$

where $w \otimes x$ is a three-dimensional convolution of a *filter* w with the $N \times M \times T$ video blocks x and $f()$ is any non-linear component-wise operation. $w \otimes x$, and as a consequence $c(x)$ also have size $N \times M \times T$. (In practice, convolution may result in size shrinking). $g()$ is a *pooling* function that results in a shrinking of the argument and operates on any $N \times M \times T$ input to generate a $J \times K \times L$ output $p(x)$, where $J \leq N$, $K \leq M$, and $L \leq M$.

4.1. Handcrafted Video Features

A typical handcrafted video feature extraction procedure is often composed of two stages of convolution and pooling. The first stage purely relies on handcrafted filters and generates descriptors from local data. The second one often uses filters learned from unsupervised methods to encode the descriptors generated from the first stage and pool them together to get global features. For example, shown in Figure 2 is a schematic description of three HOG-based IDT descriptors, each of which contains two stages of convolution and pooling (marked by dashed red and green boxes, respectively) including a total of three convolution and two pooling operations. Among them, **Conv1** uses two gradient filters as w and with:

$$f(x) = x.$$

Conv2 is an oriented soft binning, which can be approximated with w being the unit directional vectors and f being non-linear activation functions such as rectified linear unit ([15]):

$$f(x) = \max(x, 0).$$

Conv3 is KMeans-based BoW, which uses KMeans centroids as w and a softmax function ([25]) as f:

$$f_k(x) = \frac{\exp(x_k)}{\sum_j \exp(x_j)},$$

where k is the kth centroid. **Pool1** is a local sum pooling:

$$g_{x,y,t}(x) = \sum_{j,l,m \in [1,d]} x_{xj,yl,tm},$$

where d is the pool size in space and time and x, y, t are the space and time locations where $g()$ applied. **Pool2** is a global sum pooling:

$$g(x) = \sum_{x,y,t} x_{x,y,t}.$$

Using above key operators, the **IDT-HOG Net** represents the procedure of generating a KMeans-based bag of words (BoW) encoded HOG feature from stacked frames. At the first stage, the stacked frames are convolved with two gradient filters followed by 8 oriented binning filters and one spatio-temporal sum pooling. During the second stage, the descriptors from the first stage are convolved with K binning filters learned using KMeans and pooled together afterwards. The **IDT-HOF Net** and **IDT-MBH Net** represent the procedures of generating KMeans encoded HOF and MBH features, respectively, from stacked optical flows. **IDT-MBH Net** is similar to the **IDT-HOG Net** except taking optical flows as inputs instead of pixels. **IDT-HOF Net** removes **Conv1** and using 9 oriented binning filters instead of 8. Note that although we use KMeans encoding as an example, other encoding methods such as Fisher Vector and VLAD have similar procedures ([30, 25]). For simplicity, we leave out the feature detection step, which can be viewed as another convolution with binary activation function. The main strength of this pipeline is that it is computationally efficient because of the layer-wise training and does not need labels to train the feature extraction module due to the objection of reconstructing the data itself. Its limitations lie in the first stage of the structure (dashed red box) in which it uses fixed parameters and structures for different sources of data.

4.2. Comparison with CNN-based video features

Needless to say CNNs employ convolution-pooling architecture. In CNNs, the non-linear activation is generally given by $f(x) = \tanh(x)$, $f(x) = (1 + e^{-x})^{-1}$ or $f(x) = \max(x, 0)$. The pooling functions are local average or maximum pooling, for example,

$$g_{x,y,t}(x) = \max_{j,l,m \in [1,d]} x_{xj,yl,tm},$$

where d is the pool size in space and time. The parameters of the model are the filters w. These are learned by minimizing a loss function, typically defined by

$$\min_w \sum_{i=1}^{n} ||y_i - h(w, x_i)||^2$$

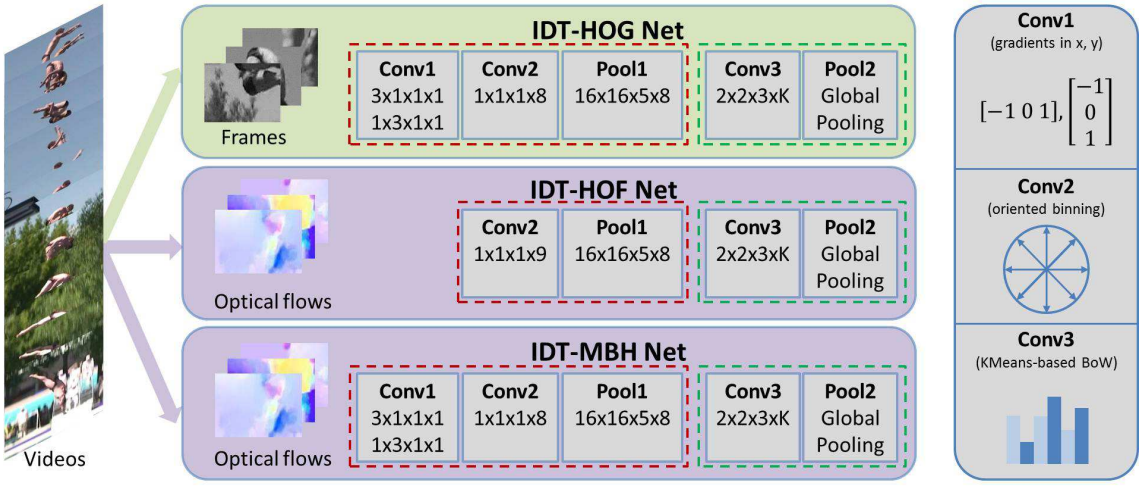


Figure 2: Schematic description of IDT as procedures of multiple convolution and pooling operations. Dashed red and green boxes represent the procedure of generating handcrafted descriptors and KMeans-based BoW encoding, respectively. In each operation, the first three numbers are the receptive field sizes in space and time (x, y, t) and the last number indicates the size of output channels.

where $h(w, \cdot)$ is the full convolution-pooling architecture that takes x as inputs. As can be noted above, the loss function requires the labels y of the training data.

Comparing the above two procedures, it is clear that their differences are not so much structural, but rather in how to get the network parameters. With this understanding, we try to answer the question of how to design a video feature learning algorithm that balances efficiency and effectiveness. At first, one might try performing end-to-end training on the network structure in Figure 2. However, this training again requires labels and large computational resources. In addition, results from [30] and [25] show that directly applying end-to-end learning on the traditional handcrafted pipelines would not bring large performance gains. So instead we keep the stage-wise unsupervised training to avoid the costly labeling and training. We address the limitations of handcrafted features by proposing a two-stream ISA-IDT to replace the handcrafted filters and enhance the proposed algorithms with two well motivated improvements.

5. Two-stream ISA-IDT

In this section, we will describe two-stream ISA-IDT in detail and its structures for both appearance (pixel) and motion (optical flow) stream learning ([27]).

As illustrated in Figure 3, an ISA ([9]) is a unsupervised feature learning method that can be described as a two-layered network within convolution-pooling architecture with: $f(x) = x^2$ and $g(x) = \sqrt{x}$. Specifically, 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 inputs and d is the dimension of

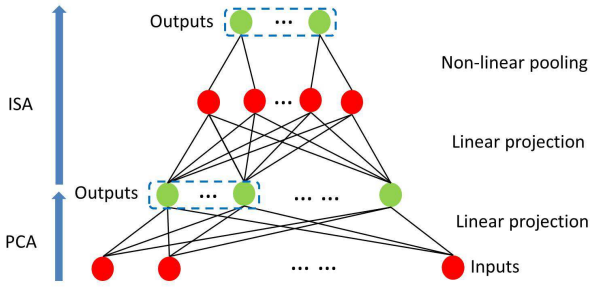
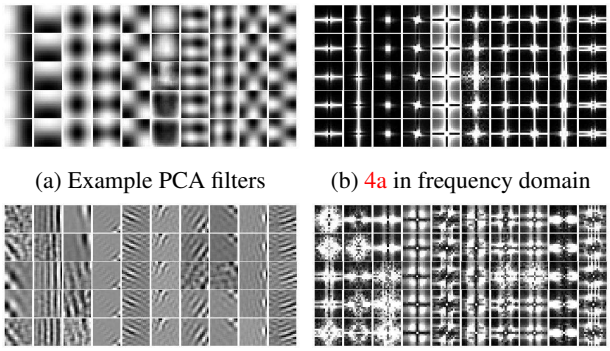


Figure 3: The neural network architecture of an ISA network with PCA preprocessing. The dashed blue boxes represent the outputs of our model.

outputs. 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

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

ISA enforces the activation of the output unit to be sparse. To achieve the sparse activation, it minimizes the following



(a) Example PCA filters (b) 4a in frequency domain

(c) Example ISA filters (d) 4c in frequency domain

Figure 4: Example filters learned from our ISA-IDT model. For all figures, y-axis represents same component at different time step and x-axis represents different components. In frequency visualization, zero-frequency component are centered in the figure.

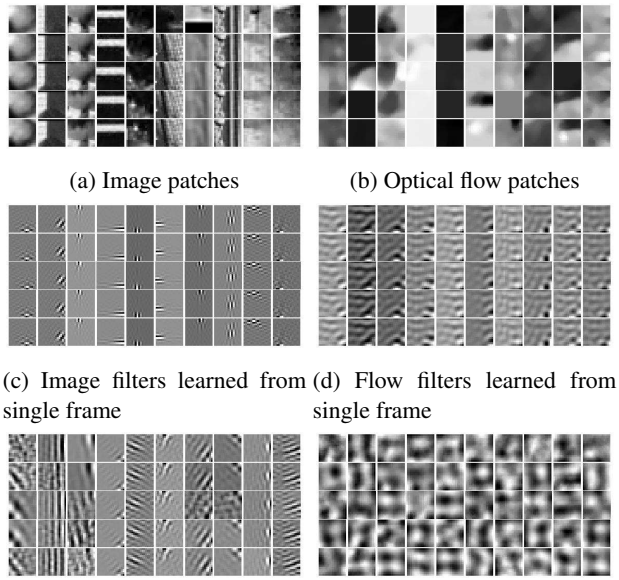
loss function defined on T training instances:

$$\min_{W} \quad \sum_{t=1}^{T}\sum_{i=1}^{d} p_i(X^t; W, V,) \qquad (2)$$
$$\text{s.t.} \quad WW^\top = I \ .$$

We select ISA as our unsupervised learning method because it is one of the best unsupervised feature learning method [9]. Also, our theoretic analysis (in the Appendix) shows that ISA is connected to the popular group Lasso algorithms.

Figure 4 visualizes, in both original and frequency domains, some example filters learned from ISA and PCA models. As illustrated in Figure 4b and 4d , the ISA model learns more complex filters that capture higher-frequency information while PCA capture lower-frequency information. To have a more complete frequency coverage, we combine the outputs of ISA with an equivalent number of top outputs from PCA. As will be shown in the experimental section, our enhanced method, denoted by ISA+, significantly outperforms individual PCA or ISA model.

To reflect the different characteristics of different data sources, we design different network structures for pixel and optical flow data. Our learned descriptor for appearance stream, dubbed LOP, is learned by directly applying a ISA+ model to a stack of video frames and implicitly learning temporal pooling. Our learned descriptor of motion stream, denoted as LOF, is trained by applying a ISA+ model to each individual optical flow frame and explicitly performing a temporal pooling afterwards. This difference of the network structures is from of our observation that pixel data has high temporal correlation while optical flow data often has much less temporal correlation due to the estimation



(a) Image patches (b) Optical flow patches

(c) Image filters learned from single frame (d) Flow filters learned from single frame

(e) Image filters learned from 5 stacked frames (f) Flow filters learned from 5 stacked frames

Figure 5: Example inputs and filters learned from our ISA models. For each figure, the y-axis represents same component at different time steps and the x-axis represents different component expect on Figure 5c and 5d, we replicate the filters 5 times for visualization purpose.

error. As a result, it is much easier to learn temporally consistent appearance filters than temporally consistent motion filters. As shown in Figure 5, in which we show some example images and optical flow patches and the filters learned in both structures. From Figure 5a and 5b, it is clear that image patches are consistent across frames while optical flow patches have large temporal variation. Quantitatively, we estimate a 0.8014 pixels correlation while only 0.2808 for optical flow correlation by using the Pearson product-moment correlation coefficients to measure their temporal correlation from 100000 random sampled HMDB51 tracklets. As a result, the learned appearance filters (Figure 5e) are temporally consistent and the learned flow filters are quite chaotic (Figure 5f). We suspect that a better optical flow will have higher correlation, but we have not explored this direction further because the Farneback optical flow we used has shown to be the best optical flow for IDT [32]. To discriminate the implicit temporal pooling of LOP from the explicit temporal pooling for optical flow data, we call it temporal projection.

6. Experiments

In the following section, we first show that our ISA+ model performs significantly better than either ISA or PCA. We then empirically demonstrate that custom designed net-

	PCA	ISA	ISA+
HMDB51	58.1%	58.4%	**61.5%**
UCF101	85.9%	86.2%	**88.3%**

Table 2: Comparison of different unsupervised feature learning methods.

work structures for pixel and optical flow data are necessary. After that, we compare our methods to the state-of-the-art video features in both descriptors and overall performance. We conduct experiments on benchmark action recognition datasets of HMDB51 and UCF101 datasets.

6.1. Datasets

The HMDB51 dataset ([16]) has 51 action classes and 6766 video clips extracted from digitized movies and YouTube. [16] provides both original videos and stabilized ones. We only use original videos in this paper. As in [16], Mean accuracy (MAcc) is used for evaluation.

The UCF101 dataset ([28]) has 101 action classes spanning over 13320 YouTube videos clips. We use the standard splits with training and testing videos provided by [28]. We report MAcc as in the original papers.

6.2. Experimental settings

As in [34], 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. Two-stream ISA-IDT models are trained on 200000 IDT trackets for each stream of data. For both PCA and ISA, we keep the filter size the same as in the handcrafted descriptors and use a pooling size of 10 for ISA. Another PCA is used to reduce the dimensionality of the resulting 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 256000 randomly sampled data points. After encoding, we attach the normalized space-time location information to the encoded descriptors as suggested in [17]. Power and ℓ_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 [34] and the one-versus-all approach is used for multi-class classification scenario. Note that we still need label for training SVM classifiers. What we try to avoid is the labels for training the feature extraction procedure, which, because of its much larger parameter size, requires a much larger number of labels.

6.3. ISA+ is better than individual PCA or ISA models

Table 2 compares our ISA+ model with individual PCA and ISA models. First, comparing PCA and ISA, we observe that, surprisingly, a simple PCA model can get similar results to a much more complex ISA model. These results demonstrate that PCA can learn good features when the number of features to generate is small. By combining the PCA and ISA outputs, we get more than 3% improvement on HMDB51 and 2% on UCF101. It should be noted that these improvements are on the combined results of appearance and motion models where the potential for improvement is smaller.

	LOG		LOF	
	Projection	Pooling	Projection	Pooling
HMDB51	**52.4%**	44.3%	46.2%	**59.5%**
UCF101	**80.0%**	73.5%	79.6%	**84.8%**

Table 3: Comparison of temporal projection and temporal pooling.

6.4. Temporal projection versus temporal pooling

In Table 3, we compare the results of temporal projection and temporal pooling. As evidenced by the results of both datasets, for appearance modeling, temporal projection is better than temporal pooling, and for motion modeling, temporal pooling performs much better than temporal projection. Furthermore, if we compare single frame image filters (Figure 5c) to filters learned using 5 frame stacks (Figure 5e) , we can see that adding temporal variation can help to learn more complex filters. A potential improvement, therefore, is to explicitly enforce temporal coherence for optical flow filter learning and learn the temporal pooling for optical flow data.

6.5. Performance comparison of individual descriptors

In Table 4, we compare our learned descriptors LOG and LOF to the video descriptors from IDT and spatial (S-CNNs) and temporal (T-CNNs) CNNs [1] from [27]. On the appearance descriptors, an impressive performance improvement of more than 10% over HOG, from 42.0% to 52.4% is achieved by LOG on HMDB51. For UCF101, LOG also gets more than 7% improvement over HOG and Spatial CNNs despite the fact that Spatial CNNs utilize additional training data. The same trend can be observed on motion descriptors. LOF outperforms other descriptors by more than 4% on HMDB51 and more than 3% on UCF101. Although it may not surprise that LOG outperforms HOG since it has been shown that unsupervised learned appearance descriptors can outperform handcrafted descriptors. However, as far as we know, we are the first to show that unsupervised motion descriptors (LOF) can outperform MBH,

[1] The first split results from [27], pretrained on Imagenet and trained HMDB51 and UCF101 together (multi-task learning)

	Appearance Descriptors				Motion Descriptors				
	HOG	S-CNNs	LOG(ConvISA)	LOG	HOF	MBH	T-CNNs	LOF(ConvISA)	LOF
HMDB51	42.0%	N.A.	47.2	**52.4%**	49.8%	52.4%	55.4%	51.0	**59.5%**
UCF101	72.4%	72.8%	79.3	**80.0%**	74.6%	81.4%	81.2%	81.2	**84.8%**

Table 4: Comparison of our proposed descriptors to IDT and two-stream CNNs.

which is currently the best handcrafted motion descriptor. On the other hand, if we simply adopt the ConvISA [19] structures that we designed for learning from pixels, we get worse results (indicated by LOF (ConvISA)) than MBH. These results again show that unsupervised optical flow feature learning is quite difficult.

6.6. Comparing with the State-of-the-Art

In Table 5, we first show that ISA-IDT can incorporate MIFS that were developed for IDT and get improved performance from that. By using MIFS, we can improve the performance of ISA-IDT on HMDB51 and UCF101 by about 3% and 1%, respectively. Second, we show that when combine the learned descriptors (ISA-IDT) with handcrafted descriptors (IDT) (Hybrid), we get further improvement on both datasets by another 2% and 1%. We also compare our performance with some state-of-the-art approaches. Note that *although we list several most recent approaches here for comparison purposes, most of them are not directly comparable to our results due to the use of different features and representations*. For IDT-ISA, the most comparable one is Wang & Schmid. [34], from which we build on our approaches. For IDT-ISA + MIFS, the most comparable one is Lan *et al.* [17], which developed and incorporated MIFS for IDT. Peng et al. [23] improved the performance of Improved Dense Trajectory by increasing the codebook size and fusing multiple coding methods. Simonyan & Zisserman [27] reported results that is competitive to IDT method by training deep convolutional neural networks using both sampled frames and optical flows and get 57.9% in HMDB51 and 87.6% in UCF101. Both Jiang *et al.* and Wu *et al.* are improvements of two-stream CNNs [27]. Jain *et al.* [10]'s result is the combination of 15000 image concepts and the results of Peng *et al.* [24], which stacks two layers of Fisher Vectors.

HMDB51		UCF101	
Oneata *et al.* [22]	54.8	Wang *et al.* [33]	85.9
Wang *et al.* [34]	57.2	Peng *et al.* [23]	87.9
Simonyan *et al.* [27]	57.9	Simonyan *et al.* [27]	87.6
Peng *et al.* [23]	61.1	Lan *et al.* [17]	89.1
Lan *et al.* [17]	65.1	Zha *et al.* [38]	89.6
Peng *et al.* [24]	66.8	Jiang *et al.* [13]	**91.1**
Jain *et al.* [10]	**71.3**	Wu *et al.* [37]	**91.3**
ISA-IDT	61.5	ISA-IDT	88.3
ISA-IDT + MIFS	64.8	ISA-IDT + MIFS	89.5
Hybrid	67.2	Hybrid	90.6

Table 5: Comparison of our results to the state-of-the-art.

7. Conclusions

Contrary to the current trend of learning video features using end-to-end deep CNNs, which is computationally demanding and label intensive, we propose in this paper to revisit the traditional local feature pipeline and combine the merits of both handcrafted and CNN approaches. As an example, we present a video feature learning algorithm that has better performance, lower computational expense than current state-of-the-art methods and does not require labels. We show that filters learned in an unsupervised fashion, when incorporated in convolution-pooling structures that are custom designed for pixel and optical flow data, can outperform supervised end-to-end networks. This result serves as a reminder that the design choices in handcrafted features may still have many useful properties which could be potentially incorporated into future deep action recognition networks. Future work would be explicitly enforcing temporal consistency for optical flow feature learning and developing a deeper and better unsupervised learning method. We would also like to explore end-to-end fine-tuning given the unsupervised learned networks, which is less expensive than training from scratch.

8. Acknowledgement

This work was partially supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: 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. The work was also supported in part by the U. S. Army Research Office (W911NF- 13-1-0277) and National Science Foundation under Grant No. IIS-1251187. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ARO and NSF.

References

[1] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733. IEEE, 2003. 2

[2] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2014. 2

[3] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, T. Tuytelaars, and L. Belgium. Modeling video evolution for action recognition. In *CVPR*, 2015. 3

[4] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010. 10

[5] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, 2014. 1

[6] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013. 1

[7] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *ACCV*, 2014. 3

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

[9] 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. 5, 6

[10] M. Jain, J. C. van Gemert, and C. G. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015. 8

[11] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, 2009. 3

[12] 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. 2

[13] Z. Jiang, Y. Wang, L. Davis, W. Andrews, and V. Rozgic. Learning discriminative features via label consistent neural network. *arXiv preprint arXiv:1602.01168*, 2016. 8

[14] 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. 1, 2, 3

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3, 4

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 7

[17] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. *CVPR*, 2015. 3, 7, 8

[18] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005. 2

[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. 2, 3, 8

[20] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 1

[21] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *arXiv preprint arXiv:1503.08909*, 2015. 1, 2

[22] D. Oneata, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013. 8

[23] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *arXiv preprint arXiv:1405.4506*, 2014. 3, 8

[24] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*. 2014. 1, 8

[25] A. Richard and J. Gall. A bow-equivalent recurrent neural network for action recognition. In *BMVC*, 2015. 3, 4, 5

[26] 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. 3

[27] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 3, 5, 7, 8

[28] 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. 7

[29] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015. 3

[30] V. Sydorov, M. Sakurada, and C. H. Lampert. Deep fisher kernels–end to end learning of the fisher kernel gmm parameters. In *CVPR*, 2014. 3, 4, 5

[31] B. Varadarajan, G. Toderici, S. Vijayanarasimhan, and A. Natsev. Efficient large scale video classification. *arXiv preprint arXiv:1505.06250*, 2015. 3

[32] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1, 2, 3, 6

[33] H. Wang and C. Schmid. Lear-inria submission for the thumos workshop. In *ICCV Workshop*, 2013. 8

[34] H. Wang, C. Schmid, et al. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 2, 3, 7, 8

[35] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009. 1

[36] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. 2015. 3

[37] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *arXiv preprint arXiv:1504.01561*, 2015. 3, 8

[38] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*, 2015. 3, 8

9. Appendix

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 $V_{i,j} = 1$. $|\mathcal{G}_i|$ defines group size, which is generally the same across groups.

As in group LASSO ([4]), for any vector $\boldsymbol{a} \in \mathbb{R}^m$, we defined the group ℓ_1-norm $\|\boldsymbol{a}\|_{\mathcal{G},1}$ as

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

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

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

Denote $\boldsymbol{\alpha}_t = WX^t$, since $WW^\top = I$, we have

$$X^t = W^\dagger \boldsymbol{\alpha}_t \, ,$$

where $W^\dagger$ is the Moore–Penrose pseudo inverse of W. Eq. (2) can be re-formulated as a sparse coding method that

$$\begin{aligned} \min_{W, \boldsymbol{\alpha_t}} \quad & \sum_{t=1}^{T} \|\boldsymbol{\alpha}_t\|_{\mathcal{G},1} \\ \text{s.t.} \quad & (W^\dagger)^\top W^\dagger = I \qquad\qquad X^t = W^\dagger \boldsymbol{\alpha}_t \end{aligned} \tag{3}$$

Based on Eq. (3), ISA is essentially searching a group-sparse representation $\boldsymbol{\alpha}_t$ of the input signal X_t. The matrix $W^\dagger$ is the dictionaries of sparse coding. The orthogonal constraint of $W^\dagger$ makes the learned components maximally independent.