Beyond Gaussian Pyramid: Multi-skip Feature Stacking for Action Recognition

Zhenzhong Lan, Ming Lin, Xuanchong Li, Alexander G. Hauptmann, Bhiksha Raj
School of Computer Science, Carnegie Mellon University
lanzhzh, minglin, xcli, alex, bhiksha@cs.cmu.edu

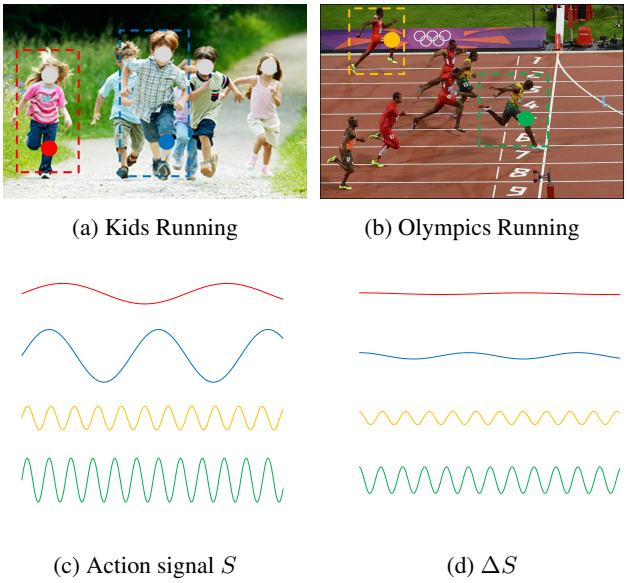
Abstract

Most state-of-the-art action feature extractors involve differential operators, which act as highpass filters and tend to attenuate low frequency action information. This attenuation introduces bias to the resulting features and generates ill-conditioned feature matrices. The Gaussian Pyramid has been used as a feature enhancing technique that encodes scale-invariant characteristics into the feature space in an attempt to deal with this attenuation. However, at the core of the Gaussian Pyramid is a convolutional smoothing operation, which makes it incapable of generating new features at coarse scales. In order to address this problem, we propose a novel feature enhancing technique called Multi-skIp Feature Stacking (MIFS), which stacks features extracted using a family of differential filters parameterized with multiple time skips and encodes shift-invariance into the frequency space. MIFS compensates for information lost from using differential operators by recapturing information at coarse scales. This recaptured information allows us to match actions at different speeds and ranges of motion. We prove that MIFS enhances the learnability of differential-based features exponentially. The resulting feature matrices from MIFS have much smaller conditional numbers and variances than those from conventional methods. Experimental results show significantly improved performance on challenging action recognition and event detection tasks. Specifically, our method exceeds the state-of-the-arts on Hollywood2, UCF101 and UCF50 datasets and is comparable to state-of-the-arts on HMDB51 and Olympics Sports datasets. MIFS can also be used as a speedup strategy for feature extraction with minimal or no accuracy cost.

1. Introduction

We consider the problem of enhancing video representations for action recognition, which becomes increasingly important for both analyzing human activity itself and as a component for more complex event analysis. As pointed out by Marr [16] and Lindeberg [14], visual representations, or visual features, are of utmost importance for a vision system, not only because they are the chief reasons for tremendous progress in the field, but also because they lead to a deeper understanding of the information processing in the human vision system. In fact, most of the qualitative improvements to visual analysis can be attributed to the introduction of improved representations, from SIFT [15] to Deep Convolutional Neural Networks [29], STIP [12] to Dense Trajectory [38] . A common characteristic of these several generations of visual features is that they all, in some way, benefit from the idea of multi-scale representation, which is generally viewed as an indiscriminately applicable tool that reliably yields an improvement in performance when applied to almost all feature extractors.

At the core of image multi-scale representation is the re-



(a) Kids Running (b) Olympics Running

(c) Action signal S (d) ΔS

Figure 1: Simplified action signals (1c) from "running" actions (1a,1b) show dramatically difference among subjects and scenes. With such dramatic differences among action signals, a differential operator with single scale is incapable of covering a full range of action frequency and tend to lose low frequency information (the red and cyan signals) .

quirement that no new detail information should be artificially found at the coarse scale of resolution [10]. Gaussian Pyramid, a unique solution based on this constraint, generates a family of images where fine-scale information is successively suppressed by Gaussian smoothing. However, in action recognition, we often desire the opposite requirements. For example, in generating action features using differential filters, we need coarse-scale features to: 1) recover the information that has been filtered out by highpass filters at fine scales, e.g., the red and cyan signals in Figure 1c are likely to be filtered out; 2) generate features at higher frequency for matching similar actions at different speeds and ranges of motion, e.g., the orange and green signals in Figure 1c. Both of these requirements cannot be satisfied with a Gaussian Pyramid representation.

In this work, we introduce a Multi-skIp Feature Stacking (MIFS) representation that works by stacking features extracted by a family of differential filters parameterized with multiple time skips (scales). Our algorithm relies on the idea that by gradually reducing the frame rate, feature extractors with differential filters can extract information about more subtle movements of actions. MIFS has several attractive properties:

- It is an indiscriminately applicable tool that can be reliably and easily adopted by any feature extractors with differential filters, like Gaussian Pyramid,
- It generates features that are shift-invariance in frequency space, hence easier to match similar actions at different speeds and ranges of motion.
- It stacks features at multiple frequencies and tends to cover a longer range of action signals compared to conventional action representations,
- It generates feature matrices that have smaller conditional numbers and variances hence stronger learnability compared to the conventional original-scale representation based on our theoretical analysis.
- It significantly improves the performance of state-of-the-art methods based on experimental results on several real-world benchmark datasets.
- It exponentially enhances the learnability of the resulting feature matrices. Therefore the required additional number of scales is logarithmic to the bandwidth of the action signals. Empirical studies show that one or two additional scales are enough to recover the information lost by differential operators. Hence the additional computational cost of MIFS is small.
- It can be used to as a feature extraction speedup strategy with minimal or no accuracy cost. As shown in our experiments, combining features extracted from videos at lower frame rates (with different time skips) performs better than features from videos at the original frame rate at the same time requires less time to process.

In the remainder of this paper, we start by providing more background information about action recognition and multi-scale presentations. We then describe MIFS in detail, followed by theoretically proving that MIFS improves the learnability of video representations exponentially. After that, an evaluation of our method is performed. Further discussions including potential improvements are given at the end.

2. Related Work

There is an extensive body of literature about action recognition; here we just mention a few relevant ones involved with state-of-the-art feature extractors and feature encoding methods. See [2] for an in-depth survey. 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 [18, 35, 38, 40, 8], especially the Dense Trajectory method proposed by Wang et al. [38, 40], together with the Fisher Vector encoding [26] yields the current state-of-the-art performances on several benchmark action recognition datasets. Peng et al. [24, 25] further improved the performance of Dense Trajectory by increasing the codebook sizes, fusing multiple coding methods and adding a stacked Fisher Vector. Some success has been reported recently using deep convolutional neural networks for action recognition in videos. Karpathy et al. [9] trained a deep convolutional neural network using 1 million weakly labeled YouTube videos and reported a moderate success on using it as a feature extractor. Simonyan & Zisserman [32] reported a result that is competitive to Improved Dense Trajectory [40] by training deep convolutional neural networks using both sampled frames and optical flows. MIFS is an indiscriminately applicable tool that can be adopted by all of above mentioned feature extractors.

Multi-scale representation [1, 14] has been very popular for most image processing tasks such as image compression, image enhancement and object recognition. A multi-scale key-point detector proposed by Lindeberg [13] and used in by Lowe [15] to detect scale invariant key points using Laplacian pyramid methods, in which Gaussian smoothing is used iteratively for each pyramid level. Simonyan & Zisserman [33] reported a significant performance improvement on Imagenet Challenge 2014 by using a multi-scale deep convolutional neural network. In video processing, Space Time Interest Points (STIP) [12] extends SIFT to the temporal domain by finding the scale invariant feature points in 3D space. Shao et al. [30] also try to achieve scale

invariance for action recognition using 3-D Laplacian pyramids and 3D Gabor filters. However, without awareness of the fundamental differences between image and video processing, [30] was not very successful when compared to the state-of-the-art methods.

For lab datasets where human poses or action templates can be reliably estimated, Dynamic Time Warping (DTP) [5], Hidden Markov Models (HMMs) [41] and Dynamic Bayesian Networks (DBNs) [23] are well studied methods for aligning actions that have speed variation. However, for noisy real-world actions, these methods have not shown themselves to be very robust.

3. Multi-skIp Feature Stacking (MIFS)

We now formalize our notation. For the present discussion a video X is just a real function of three variables:

$$X = X(x, y, t). \tag{1}$$

The normalized coordinates $(x, y, t) \in R^3$ are the Cartesian coordinates of the video space. Since we focus on the temporal domain, we omit (x, y) in further discussion and denote a video as $X(t)$. The length of the video is assumed to be normalized, that is $t \in [0, 1]$. In our model, the content of a video is generated by a linear mixture of k latent signals:

$$\bar{X} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \cdots, \bar{\mathbf{x}}_k] \,. \tag{2}$$

The mixing weight of each latent action signal $\bar{\mathbf{x}}_i$ at time t is denoted as $\alpha_i(t)$. Therefore, a given video is generated as

$$X(t) = \bar{X}\boldsymbol{\alpha}(t) + \boldsymbol{\epsilon}(t) \tag{3}$$

$$\boldsymbol{\alpha}(t) = [\alpha_1(t), \alpha_2(t), \cdots, \alpha_k(t)]^{\mathrm{T}} \,. \tag{4}$$

where $\boldsymbol{\epsilon}(t)$ is additive subgaussian noise with noise level σ. We assume $\forall i$,

$$|\alpha_i(t)| \le 1 \quad \mathrm{E}_t\{\alpha_i(t)\} = 0 \tag{5}$$

$$\mathrm{E}_t\{\alpha_i(t)^2\} \le 1 \quad \mathrm{E}_t\{\alpha_i(t) \times \alpha_j(t)|_{i \neq j}\} = 0 \,. \tag{6}$$

The feature extractor is assumed to be modeled as a differential operator $\mathcal{F}[\cdot, \tau]$ parameterized with time skip τ. Given a fixed τ, the feature extractor $\mathcal{F}[X(t), \tau]$ generates $T = \lfloor 1/\tau \rfloor$ features.

$$\mathcal{F}[X(t), \tau] = [\mathbf{f}(t_1, \tau), \mathbf{f}(t_2, \tau), \cdots, \mathbf{f}(t_T, \tau)] \,.$$

where $t_1, t_2, \cdots, t_T$ are uniformly sampled on $[0, 1]$. The i-th feature vector $\mathbf{f}(t_i, \tau)$ is generated by

$$\begin{aligned} f(t_i, \tau) &= X(t_i + \tau) - X(t_i) \\ &= \bar{X} \times (\boldsymbol{\alpha}(t_i + \tau) - \boldsymbol{\alpha}(t_i)) + \boldsymbol{\epsilon}(t_i + \tau) - \boldsymbol{\epsilon}(t_i) \,. \end{aligned} \tag{7}$$

We can rewrite the feature matrix as

$$\mathcal{F}[X(t), \tau] = \bar{X}P + \sum_{i=1}^{T} \boldsymbol{\epsilon}(t_i + \tau) - \boldsymbol{\epsilon}(t_i)$$

where P is a $k \times T$ matrix, $P_{i,j} = \boldsymbol{\alpha}_i(t_j + \tau) - \boldsymbol{\alpha}_i(t_j)$.

Most action feature extractors are different versions of $\mathcal{F}$. For example, STIP [12] and Dense Trajectory [38] can be derived from $\mathcal{F}[\cdot, \frac{1}{K}]$, where K is the number of frames in the video.

MIFS stacks multiple $\mathcal{F}[X(t), \tau]$ with different τ. By stacking multiple features with different frequencies, MIFS seeks invariance in the frequency domain via resampling in the time domain. Figure 2 shows the difference of Gaussian Pyramid and MIFS for a real signal from an unconstrained video. It is clear that, because of smoothing, Gaussian Pyramid fails to recover signals once they have been filtered out. As the levels go higher, the feature generated by Gaussian Pyramids can only become weaker. While in MIFS, the generated features become more prominent and can be recovered as the levels go higher.

4. The Learnability of MIFS

In this section, we first show that under model Eq. (2), the standard feature extraction method cannot produce a feature matrix conditioned well enough. Then we show that MIFS improves the condition number of the extracted feature matrix exponentially. One of the key novelties of the MIFS is that it also reduces the uncertainty of the feature matrix simultaneously. This reduction is not possible in a naive approach.

4.1. Condition Number of P under a Fixed τ

In this subsection, we will prove, based on the Matrix Bernstein's Inequality [37], that the condition number of P is not necessarily a small number.

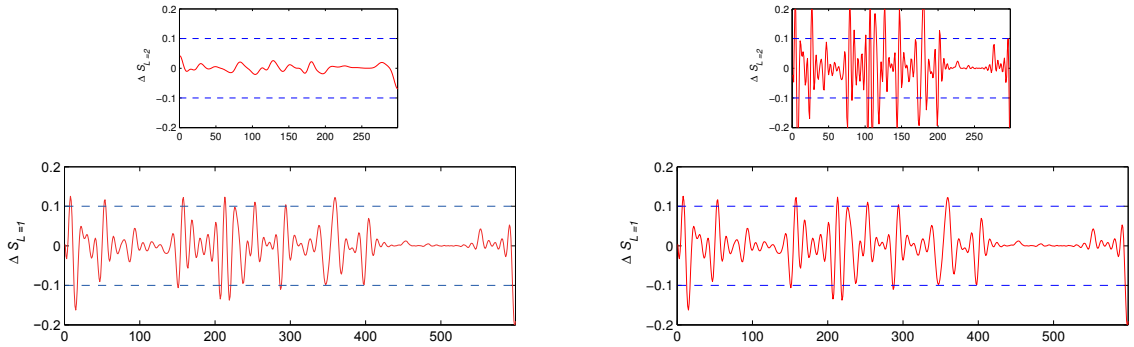
In static feature extractors such as SIFT, the weight coefficient matrix $\boldsymbol{\alpha}$ is independent of t. While in a video stream, the action signal is dynamic in t. To measure the dynamic of an action signal, we introduce γ_i as an index.

Definition 1. *A latent action signal is γ dynamic, if given a non-negative constant $c \in [0, 1]$, $\forall \tau \in (0, 1]$,*

$$1-(1+c)\exp(-\gamma/\tau) \le \mathrm{E}_t|\alpha(t)\alpha(t+\tau)| \le 1-\exp(-\gamma/\tau) \,,$$

provided $1 - (1 + c)\exp(-\gamma/\tau) \ge 0$.

The value γ measures how fast the coefficient $\alpha(t)$ varies along time t. Here we take the exponential function by assuming the correlation between $\alpha(t + \tau)$ and $\alpha(t)$ to be at least subgaussian. If in a given video, the i-th action signal is a high frequency component, then its coefficient $\boldsymbol{\alpha}_i(t)$ will behave like a random number for time skip τ.



(a) Gaussian Pyramid (b) MIFS

Figure 2: Comparison of Gaussian Pyramid and MIFS for a real action signal. The left figure (a) shows that as the level (L) goes higher (from 1 to 2), the resulting features (ΔS) from a differential operator become less prominent. So once a feature has been filtered out (assume the threshold for a feature to be represented is 0.1), it cannot be recovered by higher level features under the Gaussian Pyramid framework. The right figure (b) shows that under MIFS, the features (ΔS) become more prominent as the levels go higher and can represent those signals that have been filtered out at low levels.

Therefore, we would expect that the correlation between $\boldsymbol{\alpha}_i(t)$ and $\boldsymbol{\alpha}_i(t+\tau)$ is close to 0. Or if the action signal is a low frequency component, the correlation indicator γ should be close to 1. For the sake of simplicity, we rearrange latent action signal $\bar{X}$ by their frequency to have $\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_k$.

In a learning problem, we hope the feature matrix $\mathcal{F}[X(t), \tau]$ to be well-conditioned. Given feature matrix $\mathcal{F}[X(t), \tau]$, we can recover $\bar{X}$ by various methods, such as subspace clustering. The sampling complexity of any recovery algorithm depends on the condition number of P. Clearly when P is ill–conditioned, we require a large number of training examples to estimate $\bar{X}$. The learnability of $\mathcal{F}[X(t), \tau]$ depends on its condition number [3] which in return depends on P again. In the following, we will prove that for a fixed time skip τ, P is not necessarily well conditioned. Therefore the learnability of $\mathcal{F}[X(t), \tau]$ is suboptimal. The intuition behind our proof is that when an action signal has a large γ, then a small time skip τ will make the coefficient of that signal close to zero. Therefore, P is ill-conditioned. Formally, we have the following theorem to bound the condition number $\beta(PP^{\mathrm{T}})$ of PP^{T}(see the proof in supplementary materials).

Theorem 1. *Given a fixed time skip τ, with probability at least $1-\delta$, the condition number $\beta(PP^{\mathrm{T}})$ is bounded by*

$$\beta(PP^{\mathrm{T}}) \leq \frac{(1+c)\exp(-\gamma_1/\tau) + \Delta_\tau}{\exp(-\gamma_k/\tau) - \Delta_\tau} \tag{8}$$

$$\beta(PP^{\mathrm{T}}) \geq \frac{(1+c)\exp(-\gamma_1/\tau) - \Delta_\tau}{\exp(-\gamma_k/\tau) + \Delta_\tau} \,. \tag{9}$$

where

$$\Delta_\tau = 2\sqrt{k\frac{1}{T}(1+c)\log(2k/\delta)} \tag{10}$$

provided the number of feature points is

$$T \geq \frac{1}{9(1+c)} k \log(2k/\delta) \,. \tag{11}$$

Theorem 1 shows that when the number of features T is large enough, the condition number $\beta(PP^{\mathrm{T}})$ is a random number concentrated around its expectation $(1+c)\frac{\exp(-\gamma_1/\tau)}{\exp(-\gamma_k/\tau)}$. Since $\gamma_1 \ll \gamma_k$, the numerator is much greater than the denominator when τ is fixed. Since our proof is based on Bernstein's Inequality, the upper bound is tight. This forces $\beta(PP^{\mathrm{T}})$ to be a relatively large value. More specifically, the following corollary shows that when γ_k is linear to γ_1, $\beta(PP^{\mathrm{T}})$ is exponentially large in expectation.

Corollary 1. *When $\gamma_k \geq (M+1)\gamma_1$,*

$$\mathrm{E}\{\beta(PP^{\mathrm{T}})\} \geq (1+c)[\exp(\frac{\gamma_1}{\tau})]^M \geq (1+c)(1+\frac{\gamma_1}{\tau})^M \,. \tag{12}$$

Corollary 1 shows that when the actions in the video span across a vast dynamic range (large M), the feature extractor with single τ tends to have ill-conditioned feature matrices. A naive solution to this problem is to increase τ to reduce the condition number in expection. However, this will increase the variance Δ_τ of $\beta(PP^T)$ because of a smaller number of features. In practice, a large τ also increases the difficulty in optical flow calculation and tracking. Hence, as will also be observed in our experiments, choosing a good τ can be fairly difficult. Intuitively speaking, selecting τ is a trade-off between feature bias and variance. A feature extractor with a large τ covers a long range of action signals but with less feature points hence generates features with

small bias but large variance. Similarly, a feature extractor with a small τ will generate features with large bias but small variance.

4.2. Condition Number of P under Multiple τ

From Theorem 1, to make PP^{T} well-conditioned, we need τ as large as possible. However, when τ is too large, we cannot sample enough high quality feature points, therefore, the variance in PP^{T} will increase. To address this problem, we propose to use MIFS, which incrementally enlarges the time skip τ , then stacks all features under various τ_i to form a feature matrix. Hopefully, by increasing τ, we improve the condition number $\beta(PP^{\mathrm{T}})$ and by stacking, we sample enough features to reduce the variance.

Assuming we have features extracted from $\{\tau, 2\tau, \cdots m\tau\}$. For $i\tau$ skip, the number of extracted features is $T_i = \lfloor 1/(i\tau) \rfloor$. The following theorem bounds the condition number of MIFS (see the proof in supplementary materials).

Theorem 2. *With probability at least $1-\delta$, the condition number of PP^{T} in the MIFS is bounded by*

$$\beta(PP^{\mathrm{T}}) \leq \frac{\sum_i \frac{T_i}{T} 2(1+c)\exp(-\gamma_1/\tau_i) + \Delta_\tau}{\sum_i \frac{T_i}{T} 2\exp(-\gamma_k/\tau_i) - \Delta_\tau} . \quad (13)$$

where

$$\Delta_\tau \leq 2\sqrt{k\frac{1}{\sum_i T_i}(1+c)\log(2k/\delta)} . \quad (14)$$

Theorem 2 shows that, in the MIFS, the expected condition number $\beta(PP^{\mathrm{T}})$ is roughly the weighted average of condition numbers under various τ_i. Since $\tau_{i+1} > \tau_i$, the condition number under τ_{i+1} is smaller than the one under τ_i. Therefore, the condition number is reduced as we expected. What's nicer is that the variance component Δ_τ is actually on order of $1/\sqrt{\sum_i T_i}$, which is also much smaller than a single τ scenario. In summary, we prove:

> The MIFS representation improves the learnability of differential feature extractors because it reduces the expectation and variance of condition number $\beta(PP^{\mathrm{T}})$ simultaneously.

5. Experiments

We examine our hypothesis and the proposed MIFS representation on two tasks: action recognition and event detection. The experimental results show that MIFS representations outperform conventional original-scale representations on seven real-world challenging datasets.

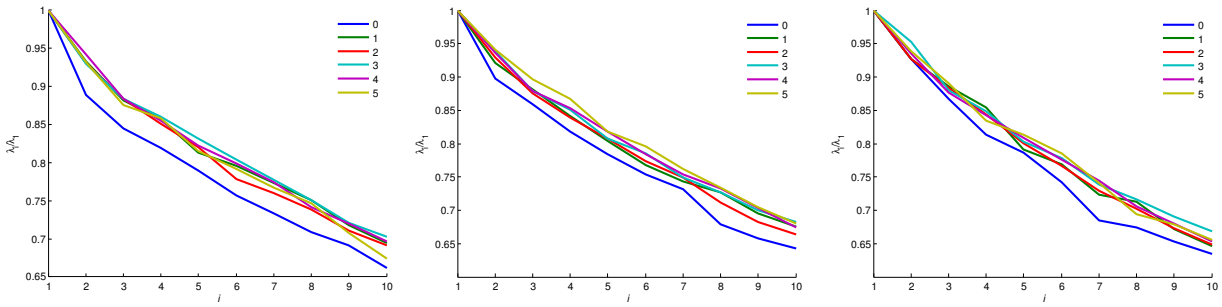
Improved Dense Trajectory with Fisher Vector encoding [40] represents the current state-of-the-arts for most real-world action recognition datasets. Therefore, we use it to evaluate our method. Note that although we use Improved Dense Trajectory, our methods can be applied to any local features that use differential filters, e.g., STIP [12].

5.1. Action Recognition

Problem Formulation The goal of this task is to recognize human actions in short clips of videos.

Datasets Five representative datasets are used: The HMDB51 dataset [11] has 51 action classes and 6766 video clips extracted from digitized movies and YouTube. [11] provides both original videos and stabilized ones. We only use original videos in this paper and standard splits with MAcc (mean accuracy) are used to evaluate the performance. The Hollywood2 dataset [17] contains 12 action classes and 1707 video clips that are collected from 69 different Hollywood movies. We use the standard splits with training and test videos provided by [17]. Mean average precision (MAP) is used to evaluate this dataset because multiple labels can be assigned to one video clip. The UCF101 dataset [34] has 101 action classes spanning over 13320 YouTube videos clips. We use the standard splits with training and test videos provided by [34] and MAcc is reported. The UCF50 dataset [27] 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 [27] is used and mean accuracy (mAcc) over all classes and all groups is reported. The Olympic Sports dataset [20] consists of 16 athletes practicing sports, represented by a total of 783 video clips. We use standard splits with 649 training clips and 134 test clips and report mAP as in [20] for comparison purposes.

Experimental Setting Improved Dense Trajectory features are extracted using 15 frame tracking, camera motion stabilization and RootSIFT normalization and described by Trajectory, HOG, HOF, MBHx and MBHy descriptors. We use PCA to reduce the dimensionality of these descriptors by a factor of two. After reduction, we augmented the descriptors with three dimensional normalized location information. The only difference between MIFS and other conventional methods is that instead of using feature points extracted from one time scale, we extract and stack all the raw feature points from different scales together before encoding. 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 L2 normalization are also used before concatenating different types of descriptors into a video based representation. Another L2 normalization is used after the concatenation. This renormalization brings us about 1% im-



(a) HMDB51 (b) Hollywood2 (c) UCF101

Figure 3: The decaying trends of singular values of feature matrices for HMDB51, Hollywood and UCF101 Datasets. 0 to 5 indicate the MIFS level and i indicates the ith singular value. From all three datasets, we can see that MIFS representations do have a slower singular value decaying trend compared to conventional representations (blue lines).

provement over the baseline method on most of the datasets except Olympics Sports. For classification, we use a linear SVM classifier with a fixed $C = 100$ as recommended by [40] and the one-versus-all approach is used for multi-class classification scenario.

Results

We first examine if it is true that the conditional number $\beta(PP^{\mathrm{T}})$ is improved by MIFS. However, it would not be meaningful to compute $\beta(PP^{\mathrm{T}})$ directly because we have noise $\boldsymbol{\epsilon}$ in $\mathcal{F}[X(t), \tau_i]$ and the smallest singular value λ_{min} is in noise space . A workaround is to examine the decaying speed of singular values of the feature matrix. The singular values are normalized by dividing the maximum singlular value $\lambda_{\max}$. We only plot the top 10 singular values, since the subspace spanned by the small singular values is noise space. Clearly, when MIFS improves the learnability, we should get a slower decaying curve of the top k singular values. Shown in Figure 3 are the trends of $\frac{\lambda_i}{\lambda_{max}}$ on the first three datasets: HMDB51, Hollywood2 and UCF101. On all three datasets, the singular values of MIFS decrease slower than the conventional one (0). It is also interesting to see that by having one or two additional levels, we have already exploited most of the potential improvement.

We further examine how performance changes with respect to the MIFS level, as shown in Table 1. First, let us compare the performance of L=0 to the standard location-insentative feature representation. Our performance on HMDB51, Hollywood2, UCF101 and UCF50 datasets are 62.1% MAcc, 67.0% MAP, 87.3% MAcc and 93.0% respectively. These numbers are higher than Wang & Schmid [40]'s results, which are 57.2%, 64.3%, 85.9% and 91.2%, respectively. This improvement is largely because of our location sensitive feature representation and the renormalization. Next, let us check the behavior of MIFS. For completeness, we list both single-scale and stacking performance. For single-scale performance, we observe that for HMDB51, its performance increases from 62.1% to 63.1% and then decrease rapidly, similar patterns can be seen in other datasets except some of them do not increase at $L = 1$. These results consist with our observation that different actions need different scale ranges. They also substantiate our proof that selecting time interval τ is a trade-off between the feature bias and its variance. If computational cost is critical, then we can choose to only extract higher single scale features but suffering minimal or no accuracy lost and enjoying large computational reduction. Now let us compare MIFS with single-scale representation. We observe that for MIFS representations, although there is still a bias and variance trade-off as in single-scale representations for different levels, they all perform better than single-scale representation and the performance decreasing points are later than those in the single-scale representations. We also observe that for MIFS representations, most of the performance improvement comes from $L = 1$ and $L = 2$, which supports what we observed in Figure 3 that, in practice, having one or two more scales is enough to recover most of lost information due to the differential operations. Higher scale features become less reliable due to the increasing difficulty in optical flow estimation and tracking. It is also interesting to observe that HMDB51 enjoys a higher performance improvement from MIFS than the other four datasets have. We believe that the main reason is that HMDB dataset is a mixture of videos from two sources: Youtube and movie, which results in larger action velocity range than pure movie videos or Youtube in other datasets.

Comparing with State-of-the-Arts In Table 2, we compare MIFS at $L = 3$, which performs well across all action datasets, with the state-of-the-art approaches. From Table 2, in most of the datasets, we observe improvement over the state-of-the-arts except hmdb51 and Olympics Sports,

	HMDB51 (MAcc%)		Hollywood2 (MAP%)		UCF101 (MAcc%)		UCF50 (MAcc%)		Olympics Sports (MAP%)	
L	single-scale	MIFS	single-scale	MIFS	single-scale	MIFS	single-scale	MIFS	single-scale	MIFS
0	62.1		67.0		87.3		93.0		89.8	
1	63.1	63.8	66.4	67.5	87.3	88.1	93.3	94.0	89.4	**92.9**
2	54.3	64.4	62.5	67.9	85.5	88.8	92.2	94.1	88.1	91.7
3	43.8	65.1	60.5	**68.0**	81.3	**89.1**	89.7	**94.4**	85.3	91.4
4	24.1	**65.4**	58.1	67.4	74.6	**89.1**	84.3	**94.4**	85.0	90.3
5	15.9	**65.4**	54.4	67.1	66.7	89.0	76.7	94.3	82.3	91.3

Table 1: Comparison of different scale levels for MIFS.

HMDB51 (MAcc. %)		Hollywood2 (MAP %)		UCF101(MAcc. %)		UCF50 (MAcc. %)		Olympics Sports (MAP %)	
Oneata *et al.* [21]	54.8	Lin *et al.* [36]	48.1	Karpathy *et al.* [9]	65.4	Shi *et al.* [31]	83.3	Jain *et al.* [7]	83.2
Wang *et al.* [40]	57.2	Sapienz *et al.* [28]	59.6	Sapienz *et al.* [28]	82.3	Sanath *et al.* [19]	89.4	Adrien *et al.* [6]	85.5
Simonyan *et al.* [32]	57.9	Jain *et al.* [7]	62.5	Wang *et al.* [39]	85.9	Arridhana *et al.* [4]	90.0	Oneata *et al.* [21]	89.0
Peng *et al.* [24]	61.1	Oneata *et al.* [21]	63.3	Simonyan *et al.* [32]	87.6	Oneata *et al.* [21]	90.0	Wang & Schmid [40]	91.1
Peng *et al.* [25]	**66.8**	Wang *et al.* [40]	64.3	Peng *et al.* [24]	87.9	Wang & Schmid [40]	91.2	Peng *et al.* [25]	**93.8**
MIFS (L=3)	65.1	MIFS (L = 3)	**68.0**	MIFS (L = 3)	**89.1**	MIFS (L=3)	**94.4**	MIFS (L = 3)	91.4

Table 2: Comparison of our results to the state-of-the-arts.

on which our $L = 3$ MIFS give inferior performance . 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*. The most comparable one is Wang & Schmid. [40], from which we build our approaches on. Sapienz *et al.* [28] explored ways to subsample and generate vocabularies for Dense Trajectory features. Jain et al. [7]'s approach incorporated a new motion descriptor. Oneata et al. [21] focused on testing Spatial Fisher Vector for multiple action and event tasks. Peng et al. [24] improved the performance of Improved Dense Trajectory by increasing the codebook size and fusing multiple coding methods. Karpathy et al. [9] trained a deep convolutional neural network using 1 million weakly labeled YouTube videos and reported 65.4% mean accuracy on UCF101 datasets. Simonyan & Zisserman [32] reported results that are competitive to Improved Dense Trajectory by training deep convolutional neural networks using both sampled frames and optical flows and get 57.9% MAcc in HMDB51 and 87.6% MAcc in UCF101, which are comparable to the results of Wang & Schmid. Peng et al. [25] achieves better results than us on HMDB51 and Olympic Sports datasets by combining a hierarchical Fisher Vector with the original one.

5.2. Event Detection

Problem Formulation Given a collection of videos, the goal of an event detection task is to detect events of interest such as *Birthday Party* and *Parade*, solely based on the video content. The task is very challenging due to complex actions and scenes. By evaluating on this task, we examine whether MIFS can improve the performance of recognizing very complex actions.

Dataset TREC Video Retrieval Evaluation (TRECVID) Multimedia Event Detection (MED) [22] is a task organized by NIST (National Institute of Standards and Technology) aimed at encouraging new technologies for detecting complex events such as *having a birthday party*. Started in 2010, NIST has gradually built up a database that contains 8000 hours of videos and 40 events, which is by far the largest event detection collection. MEDTEST13, 14 datasets are two standard system evaluation datasets released by NIST in 2013 and 2014, respectively. Each of them contains around 10 percent of the whole MED collection and has 20 events. They consist of two tasks, i.e. EK100 and EK10. EK100 task has 100 positive training samples while EK10 has 10. For both tasks, they have around 5000 background samples. Together, each dataset has 8000 training samples and 24000 testing samples.

Experimental Setting A similar setting discussed in section 5.1 is applied except we use five folders cross-validation to choose the penalty parameter C for linear SVM. For each classifier, C is chosen among $10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}$. We only test MIFS with $L = 3$ as recommended in section 5.1 because extracting Dense Trajectory feature from such large datasets itself is very time consuming. It took us 4 days to generate representations for both MEDTEST13, 14 using a cluster with more than 500 Intel E565+ series processors. We use MAP as evaluation criteria.

	HMDB51 (MAcc%)	Hollywood2 (MAP%)	UCF101 (MAcc%)	UCF50 (MAcc%)	Olympics Sports (MAP%)	Computational Cost (Relative)
L=0	62.1	67.0	87.3	93.0	89.8	1.0
L=1-0	63.1	66.4	87.3	93.3	89.4	0.5
L=2-0	63.9	67.6	88.5	93.8	91.9	0.75

Table 3: Performance versus relative computational cost for feature extraction

	MEDTEST13		MEDTEST14	
	EK100	EK10	EK100	EK10
Baseline	34.2	17.7	27.3	12.7
MIFS (L=3)	**36.3**	**19.3**	**29.0**	**14.9**

Table 4: Performance Comparison on the MED task.

Results Table 4 lists the overall MAP (detail results can be found in supplementary materials). The baseline method is a conventional single-scale representation with $L = 0$. From Table 4, we can see that for both MEDTEST13 and MEDTEST14, MIFS representations consistently improve over the original-scale representation by about 2% in both EK100 and EK10. It is worth emphasizing that MED is such a challenging task that 2% of absolute performance improvement is quite significant.

5.3. Computational Complexity

Level 0 of a MIFS representation has the same cost as other single pass methods, e.g., Wang & Schmid. [40]. For level l, the cost becomes $1/l$ of the level 0. So with a MIFS up to level 2, the computational cost will be less than twice the cost of a single pass through the original video, yet it can significantly improve the single-pass methods. If computational efficiency is critical, the method can be sped up by removing low-scale features. For example, removing L=0 (original videos) will significantly reduce cost but still give useful improvements as shown in Table 3. $L - 1$ shows the results of only using features from every 2nd frame and $L = 2 - 0$ shows the results of combining features from level 1 (every 2nd frame) and level 2 (every 3rd frame) but not L=0. As seen, in most of cases, we can still get better results with less cost.

6. Conclusion

We develop the Multi-skIp Feature Stacking (MIFS) method for enhancing the learnability of action representations. MIFS stacks features extracted using a family of differential filters parameterized with multiple time skips and achieves shift-invariance in the frequency space. In contrast to Gaussian Pyramid, MIFS generates features at all scales and tends to cover a longer range of action signals. Theoretical results show that MIFS improves the learnability of action representation exponentially. Extensive experiments on seven real-world datasets show that MIFS exceeds state-of-the-art methods. Future works would be determining the appropriate level for different action types. Additionally, we would like to improve the quality of optical flow calculation and tracking at coarse scales.

7. 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). 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.

References

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 2

[2] J. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011. 2

[3] W. Chu and T. Y. Lin. *Foundations and advances in data mining*, volume 180. Springer, 2005. 4

[4] A. Ciptadi, M. S. Goodwin, and J. M. Rehg. Movement pattern histogram for action recognition and retrieval. In *ECCV*. 2014. 7

[5] T. Darrell and A. Pentland. Space-time gestures. In *CVPR*, 1993. 3

[6] A. Gaidon, Z. Harchaoui, and C. Schmid. Activity representation with motion hierarchies. *International Journal of Computer Vision*, 107(3):219–238, 2014. 7

[7] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013. 7

[8] 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

[9] 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. 2, 7

[10] J. J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 2

[11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 5

[12] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005. 1, 2, 3, 5

[13] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *IJCV*, 11(3):283–318, 1993. 2

[14] T. Lindeberg and B. M. ter Haar Romeny. *Linear scale-space I: Basic theory*. Springer, 1994. 1, 2

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2

[16] D. Marr. Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. *Inc., New York, NY*, pages 2–46, 1982. 1

[17] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 5

[18] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Workshops*, 2009. 2

[19] S. Narayan and K. R. Ramakrishnan. A cause and effect analysis of motion trajectories for modeling actions. In *CVPR*, 2014. 7

[20] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*. 2010. 5

[21] D. Oneata, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013. 7

[22] P. Over, G. Awad, J. Fiscus, and G. Sanders. Trecvid 2013–an introduction to the goals, tasks, data, evaluation mechanisms, and metrics. 2013. 7

[23] S. Park and J. K. Aggarwal. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia systems*, 10(2):164–179, 2004. 3

[24] 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. 2, 7

[25] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *Computer Vision–ECCV 2014*, pages 581–595. Springer, 2014. 2, 7

[26] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. 2010. 2

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

[28] 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. 7

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1

[30] L. Shao, X. Zhen, D. Tao, and X. Li. Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics*, pages 2168–2267, 2013. 2, 3

[31] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *CVPR*, 2013. 7

[32] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 2, 7

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[34] 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. 5

[35] 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. 2

[36] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, and S. Yan. Dl-sfa: Deeply-learned slow feature analysis for action recognition. In *CVPR*, 2014. 7

[37] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. 3

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

[39] H. Wang and C. Schmid. Lear-inria submission for the thumos workshop. In *ICCV Workshop*, 2013. 7

[40] H. Wang, C. Schmid, et al. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 5, 6, 7, 8

[41] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, 1992. 3