Video Activity Recognition Using Sequence Kernel Based Support Vector Machines

Sony S. Allappa[1], Veena Thenkanidiyoor[1(✉)], and Dileep Aroor Dinesh[2]

[1] National Institute of Technology Goa, Farmagudi, India
sonynitgoa@gmail.com, veenat@nitgoa.ac.in
[2] Indian Institute of Technology Mandi, Mandi, Himachal Pradesh, India
addileep@iitmandi.ac.in

Abstract. This paper addresses issues in performing video activity recognition using support vector machines (SVMs). The videos comprise of sequence of sub-activities where a sub-activity correspond to a segment of video. For building activity recognizer, each segment is encoded into a feature vector. Hence a video is represented as a sequence of feature vectors. In this work, we propose to explore GMM-based encoding scheme ot encode a video segment into bag-of-visual-word vector representation. We also propose to use Fisher score vector as an encoded representation for a video segment. For building SVM-based activity recognizer, it is necessary to use suitable kernel that match sequences of feature vectors. Such kernels are called sequence kernels. In this work, we propose different sequence kernels like modified time flexible kernel, segment level pyramid match kernel, segment level probability sequence kernel and segment level Fisher kernel for matching videos when segments are represented using an encoded feature vector representation. The effectiveness of the proposed sequence kernels in the SVM- based activity recognition are studied using benchmark datasets.

Keywords: Video activity recognition
Gaussian mixture Model based encoding · Fisher score vector
Support evctor machine · Time flexible kernel
Modified time flexible kernel · Segment level pyramid match kernel
Segment level probability sequence kernel · Segment level Fisher kernel

1 Introduction

Video activity recognition is one of the most interesting tasks, due to its benefits in areas such as intelligent video surveillance, automatic cinematography, elderly behavioral management, human-computer interaction, etc. Activity recognition involves assigning an activity label to a video which human beings are good at doing. However, video activity recognition is a challenging task for a computer. This is because, a video activity comprises a sequence of sub-activities. The order

M. De Marsico et al. (Eds.): ICPRAM 2018, LNCS 11351, pp. 164–185, 2019.
https://doi.org/10.1007/978-3-030-05499-1_9

in which the sub-activities appear characterizes an activity class. For example, the activities "Getting Out of Car" and "Getting Into Car" have common set of sub-activities like 'person opens car door', 'goes out of the car', 'closes door' and 'walks'. If the sequence of sub-activities are in the order "person opens car door, goes out of the car, closes door and walks", then the activity is "Getting Out of Car". If the sub-activities are in the order "person walks, opens car door, gets into the car and closes door", the activity indicates "Getting Into Car". So, the temporal ordering of sub-activities is important for discrimination between the video activities. An automatic approach to video activity recognition should consider the temporal ordering of sub-activities to discriminate one video activity from another.

A video, which is a sequence of frames can be viewed as a three dimensional matrix that carries rich spatio-temporal information. An activity recognizer should use this rich spatio-temporal information [16–18]. This is possible by extracting suitable features that capture spatio-temporal information. This leads to representing a video as a sequence of feature vectors, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_t, .., \mathbf{x}_T)$ where, $\mathbf{x}_t \in R^d$ and T is the length of the sequence. The length of the video sequences vary from one sequence to other because videos are of different lengths. Hence, video activity recognition involves classification of varying length sequences of feature vectors. The process of video activity recognition is illustrated in Fig. 1. It is seen from Fig. 1 that, an activity recognizer involves first extracting spatio-temporal features from a video and then using a suitable classification model to recognize the activity.

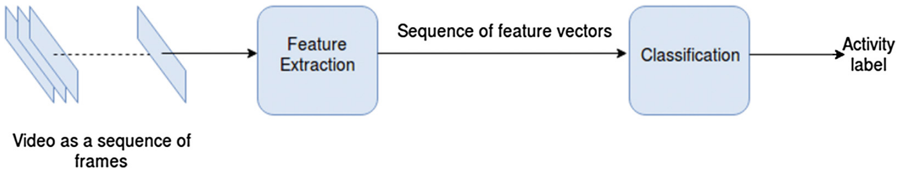

Fig. 1. Illustration of video activity recognition process.

It is necessary for a video activity recognizer to discriminate among activity classes that exhibit strong inter-class similarity. Conventionally hidden Markov model (HMM) based classifiers are used for classification of varying length sequences of feature vectors [5]. Since HMMs are built using non-discriminative learning based approaches, they may not be effective for activity recognition. Since activity classes exhibit strong inter-class similarity, a discriminative learning based approach such as a support vector machine (SVM) based classifier is expected to be helpful in building an activity recognizer. In this work, we propose to build a SVM-based classifier for activity recognition that involves classification of varying length sequences corresponding to videos.

Classification of varying length sequences of feature vectors using a SVM-based classifier requires the design of a suitable kernel as a measure of similarity between a pair of sequences. Kernels designed for varying length sequences of feature vectors are called sequence kernels [33]. In this work, we propose the design of sequence kernels for video activity recognition. While designing sequence kernels for video activity recognition, it is necessary to consider the temporal ordering of sub-activities. A sub-activity in a video corresponds to a small portion of video called a segment of the video. This segment of a video brings the contextual information that is necessary for activity recognition. So, a sequence of sub-activities correspond to sequence of segments. It is now necessary that in the design of a sequence kernel the context segments of videos should be considered. However, segmenting a video into sub-activities is not a trivial task. Instead, in this work we propose to consider the contextual segments in different ways for designing the sequence kernels. These contextual segments are approximately corresponding to sub-activities. In this work, we propose to encode a context segment of a video in two ways and then this encoded representation is used in the kernel computation. In the first approach to encoding, a segment of a video is encoded into a bag-of-visual-words (BOVW) vector. The visual words represent local semantic concepts and are obtained by clustering the feature vectors of all the video sequences of all the activity classes. The BOVW vector corresponds to histogram of visual words that occur in that segment. In the second approach, a segment of video is encoded into a Fisher score vector obtained using Gaussian mixture model (GMM) based likelihood scores. This involves first building a GMM using the feature vectors of all the video sequences of all the activity classes. A Fisher score vector is obtained using the first order derivatives of the log-likelihood with respect to the parameters of a GMM.

In this work, we extended the previous work of [36] and propose to compute time flexible kernel (TFK) and modified time flexible kernel (MTFK) where, each context segment is encoded into Fisher score vector. To design TFK [1], a video divided into segments of a fixed number of frames and every segment is encoded into a BOVW vector. As a result, a video is represented as a sequence of BOVW vectors. Matching a pair of videos using TFK involves matching every BOVW vector from a sequence with every BOVW vector from other sequence. In TFK, a pair of BOVW vectors is matched using a linear kernel (LK). In the design of MTFK [36], better approaches to match a pair of BOVW vectors are considered. The BOVW representation corresponds to frequency of occurrence of visual words. It is shown in [6] that frequency based kernels are suitable for matching a pair of frequency based vectors. In the design of MTFK, the frequency based kernels are used for matching a pair of BOVW vectors [36]. In this work, we propose to encode every segment into Fisher score vector. As a result, a video is represented as a sequence of Fisher score vectors. We then propose to compute TFK using Improved Fisher kernel (IFK) to match a pair of Fisher score vectors.

It is possible for a sequence of sub-activities at finer level to correspond to higher level sub-activities. For example, in the game of cricket, ‘bowling event’ may comprise of ‘running’ and ‘throwing a ball’. Hence matching a pair of video sequences at different abstract levels of sub-activities may be helpful. We propose to explore segment level pyramid match kernel (SLPMK) [21] and segment level probabilistic sequence kernel (SLPSK) [32], used in the context of speech for activity recognition where matching between a pair of videos is done at different levels of segments. The design of SLPMK is inspired by the spatial pyramid match kernel [21]. Here, a video sequence is decomposed into pyramid of increasingly finer segments. Every segment is encoded into a BOVW vector. The SLPMK between a pair of video sequences is computed by matching the corresponding segments at each level in the pyramid. In the design of SLPSK, a video is divided into a fixed number of segments. Every segment is mapped onto a high dimensional probabilistic score space. The proposed SLPSK is computed as a combination of probabilistic sequence kernel (PSK) computed between a pair of segments which corresponds to inner product between the probabilistic score space representation.

Inspired by the concept of SLPMK, we propose segment level Fisher kernel (SLFK) for video activity recognition. In the design of SLFK, every segment is encoded as a Fisher score vector and the SLFK is computed as a combination of IFK computed between the segments of the videos. The effectiveness of the proposed kernels is studied using benchmark datasets.

The main contributions of this paper are as follows.

$*$ GMM-based encoding scheme to encode a segment of a video into a BOVW vector representation. The GMM-based approach uses soft assignment to clusters which is found to be effective when compared to codebook based encoding which uses K-means clustering.

$*$ Fisher score vector, as an encoded representation for a context segment of video.

$*$ TFK and MTFK where each segment is encoded into BOVW vector representation and Fisher score vector representation.

$*$ SLPMK, SLPSK and SLFK for matching a pair of videos at different levels of segments (or sub-activities).

$*$ Demonstration of the effectiveness of proposed sequence kernel based SVM, with state-of-the-art results, for video activity recognition.

This paper is organized as follows. A brief overview of approaches to activity recognition is presented in Sect. 2. In Sect. 3, we present video representation techniques. Dynamic kernels used in this work are presented in Sect. 4. The experimental studies are presented in Sect. 4. In Sect. 5, we present the conclusions.

2 Video Representation

For building an effective activity recognizer, it is important to represent a video in a suitable way. This requires first to extract spatio-temporal features from a video. The sequential kernels proposed in this work need to consider matching a pair of videos using segments of videos to consider temporal ordering of sub-activities. Hence, it is useful to encode a video segment into a suitable representation. In this section, we first present an approach to feature extraction and then we present approaches to encode a video.

Video has a rich spatio-temporal information. The feature descriptors should be such that, they preserve the spatio-temporal information. Conventional approaches to video representation were extended from image representation that involves extracting features such as edges, colors, corners, etc. from every frame of video [16,18]. In this method, every frame is represented by a feature vector so that a video is represented as a sequence of feature vectors to capture temporal information. In this work, we propose to consider improved dense trajectories (IDT) based features. The IDT is a state-of-the-art descriptor, used to retain both spatial and temporal information effectively. An illustration for IDT-based feature extraction is given in Fig. 2. As shown in Fig. 2, the process densely samples feature points in each frame and tracks them in the video based on optical flow. Instead of mapping an individual frame, or a group of frames sequentially, this approach uses the sliding window method to keep the temporal information intact. We choose a window size of B frames, and sliding length of F frames. Multiple descriptors are computed along the trajectories of feature points to capture shape, appearance and motion information. The descriptors such as Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF) [10] and Motion Boundary Histograms (MBH) are extracted from each trajectories. The IDT descriptor is found to be effective for video analysis tasks [1]. Capturing spatio-temporal information in videos leads to representing a video as a sequence of feature vectors $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t, ..., \mathbf{x}_T)$ where $\mathbf{x}_t \in R^d$ and T is the length of the sequence. Since videos are of different lengths, the lengths of the corresponding sequences of feature vectors also vary from one video to another.

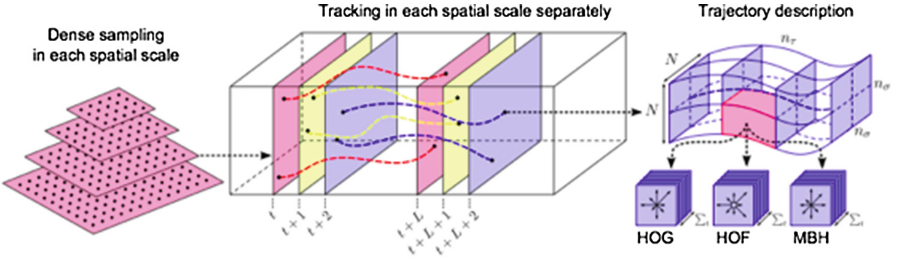

Fig. 2. An illustration for the process of improved dense trajectories (IDT) based feature extraction [10].

An activity is a sequence of sub-activities. A sub-activity corresponds to a small portion of a video called segment. It would be helpful to represent a video in terms of the segments. In this work, we propose to encode a segment of a video into a suitable representation and then design the sequence kernel using them. We propose to consider two methods for video encoding. The first method involves encoding a video segment into bag-of-visual-words (BOVW) representation. The second method encodes a segment of the video into a Fisher score vector.

BOVW Encoding: The BOVW representation of a video, $\mathbf{X}$ corresponds to a vector of frequencies of occurrence of visual words denoted by $\mathbf{z} = [z_1, z_2, ...z_k, ..., z_K]^T$. Here, z_k corresponds to the frequency of occurrence of the k^{th} visual word in the video and K is the total number of visual words. A visual word represents a specific semantic pattern shared by a group of low-level descriptors. To obtain BOVW representation, every feature vector is assigned to the closest visual word. Visual words are obtained by clustering all the feature vectors $\mathbf{x} \in R^d$ from all the videos into K clusters. Conventionally K-means clustering approach is used to cluster the feature vectors of all the video sequences. The visual words correspond to the cluster centers and are also called as codewords. An encoding approach using a set of codewords is known as codebook based encoding (CBE). K-means clustering is a hard clustering approach. Soft cluster assignment is expected to be better in the encoding process. In this work, we propose to consider Gaussian mixture model (GMM) based approach for soft clustering. This involves building a GMM and every component of GMM is considered as a representation for a visual word. The video encoding using a GMM is called as GMM-based encoding (GMME). The CBE and GMME for video encoding that are presented below.

Codebook Based Encoding: In the codebook based encoding (CBE) scheme a feature vector is assigned with the index of the closest visual word using Euclidean distance as follows:

$$m = \underset{k}{\operatorname{argmin}} ||\mathbf{x} - \boldsymbol{\mu}_k|| \tag{1}$$

Here, $\boldsymbol{\mu}_k$ denotes the k^{th} visual word that corresponds to the center of k^{th} cluster. In this method only the centers of the clusters are used. Information such as spread of the cluster and the strength of the cluster are not considered.

Gaussian Mixture Model Based Encoding: In this method, a feature vector can belong to more than one cluster with non-zero probability. In this method, components of the GMM correspond to the visual words. The belongingness of a feature vector $\mathbf{x}$ to a cluster k in the GMM-based encoding (GMME), is given by the responsibility term,

$$\gamma_k(\mathbf{x}) = \frac{w_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{C}_k)}{\sum_{i=1}^{K} w_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{C}_i)} \tag{2}$$

where w_k is the mixture coefficient of the component k, and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{C}_k)$ is the Gaussian density for the component k with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{C}_k$. For a video sequence with T feature vectors, the z_k is computed as,

$$z_k = \sum_{t=1}^{T} \gamma_k(\mathbf{x}_t) \tag{3}$$

Our studies and experimental results show that the performance obtained for activity recognizer that uses GMME for representing a video is better compared to that uses CBE method. In the encoding process presented so far, entire video is encoded into a BOVW representation that corresponds to a histogram of visual words. An important limitation of encoding entire video, $\mathbf{X}$ into $\mathbf{z}$ is that the temporal information among the frames of the video is lost. As video activity is a very complex phenomena that involves various sub-activities, the temporal information corresponding to the sub-activities in an activity is essential for video activity recognition. The sub-activities also correspond to segments of a video and the ordering of these segments is important. The segments bring contextual information. To build an activity recognizer, it is helpful to match a pair of videos at segment level. For this each segment can be encoded into a BOVW vector. This requires to split a video into a sequence of segments and encode every segment into a BOVW vector. This results in a sequence of BOVW vectors representation for a video, $\mathbf{Y} = (\mathbf{y}_{\ 1}, \mathbf{y}_{\ 2}, ..., \mathbf{y}_{\ n}, ..., \mathbf{y}_{\ N})$ where $\mathbf{y_n} \in R^K$. Here, N corresponds to the number of video segments considered.

Fisher Encoding: In this work, we also propose to encode a video segment as a Fisher score vector.

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..\mathbf{x}_t, ...\mathbf{x}_T)$, where $\mathbf{x}_t \in R^d$ denote a segment of a video. Let γ_{kt}, $k = 1, \ldots, K, t = 1, \ldots, T$ denote the soft assignments of the T feature vectors of a video segment to K Gaussian components. For each $k = 1, \ldots, K$, define the vectors

$$\mathbf{u}_k = \frac{1}{N\sqrt{w_k}} \sum_{t=1}^{T} \gamma_{kt} \mathbf{C}_k^{-1/2} (\mathbf{x}_t - \boldsymbol{\mu}_k) \tag{4}$$

$$\mathbf{v}_k = \frac{1}{N\sqrt{2w_k}} \sum_{t=1}^{T} \gamma_{kt} [(\mathbf{x}_t - \boldsymbol{\mu}_t) \mathbf{C}_k^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_k) - 1] \tag{5}$$

The Fisher encoding of the sequence of feature vector is then given by the concatenation of $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ for all K components.

$$\mathbf{f} = [\mathbf{u}_1^T, \mathbf{v}_1^T, \ldots, \mathbf{u}_K^T, \mathbf{v}_K^T]^T \tag{6}$$

To take into consideration the temporal ordering of video sub-activities, we propose to encode an activity video into a sequence of Fisher vectors as $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, ...\mathbf{f}_n, ..., \mathbf{f}_N)$.

Video activity recognition is a challenging task due to the high level of interclass similarity exhibited by the activity classes. It is very essential for an activity recognizer to discriminate among the activity classes. Support vector machines (SVMs) are shown to be effective in building discriminative classifiers. For building an effective activity recognizer using SVMs, it is necessary to use suitable kernels. We present the kernels proposed in this work in the next section.

3 Sequence Kernels for Video Activity Recognition

Activity recognition in videos involves considering the sequence of feature vectors $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_t, ...\mathbf{x}_T)$ representation of videos. Here, $\mathbf{x}_t \in \mathbf{R}^d$ and $\mathbf{T}$ is the length of the sequence. Videos being different in length, the length of the sequence varies from one video to another. To build an SVM-based activity recognizer, it is necessary to consider suitable kernels that consider varying length sequences of feature vectors. Kernels that consider varying length sequences of feature vectors are known as sequence kernels [33]. An activity video is a sequence of sub-activities. The temporal ordering of sub-activities is important for activity recognition. A sub-activity corresponds to a small portion of a video called segment. The segments bring contextual information for building an activity recognizer. We propose to design sequence kernels that consider these contextual segments and match a pair of video at the segment level. In this section, we propose the design of five sequence kernels namely, time flexible kernel (TFK), modified time flexible kernel (MTFK), segment level pyramid match kernel (SLPMK), segment level probabilistic sequence kernel (SLPSK) and segment level Fisher kernel (SLFK).

3.1 Time Flexible Kernel

To design a time flexible kernel (TFK), a sequence of feature vectors, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t, ..., \mathbf{x}_T)$ is first divided into sequence of segments such that, $\mathbf{X} = (\mathbf{X}^1, \mathbf{X}^2, ..., \mathbf{X}^n, ..., \mathbf{X}^N)$, where, $\mathbf{X}^n$ corresponds to n^{th} segment and N is the number of segments. A segment is obtained by considering a sliding window of $'B'$ frames. Every segment $\mathbf{X}^n$ is encoded into a bag-of-visual-word (BOVW) vector $\mathbf{y}_n \in R^K$ where, $\mathbf{y}_n = [y_{n1}, y_{n2}, .., y_{nk}, .., y_{nK}]^T$. This results in encoding a sequence of feature vectors, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t, ..., \mathbf{x}_T)$ into a sequence of BOVW vectors, $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_n, .., \mathbf{y}_N)$.

Let $\mathbf{Y}_i = (\mathbf{y}_{i1}, \mathbf{y}_{i2}, ..., \mathbf{y}_{in}, ..., \mathbf{y}_{iN})$ and $\mathbf{Y}_j = (\mathbf{y}_{j1}, \mathbf{y}_{j2}, ..., \mathbf{y}_{jm}, ..., \mathbf{y}_{jM})$ correspond to the sequence of BOVW vectors representation of i^{th} and j^{th} videos respectively. Here, N and M represent the length of the sequence of BOVW vectors. The TFK involves matching every BOVW vector from $\mathbf{Y}_i$ with every BOVW vector from $\mathbf{Y}_j$. It was observed that in an activity video, middle of the video has the core of the activity [1] and it is necessary to ensure maximum matching at the middle of the video. Hence to effectively match a pair of videos, it is necessary to ensure that their centers are alligned. This is achieved by using a weight, w_{nm} for matching between y_{in} and y_{jm}. The value of w_{nm} is large when $n = N/2$ and $m = M/2$, i.e., matching at the center of two sequences. The value of w_{nm} for $n = N/2, m = 1$ will be smaller than w_{nm} for $n = N/2, m = M/2$. The details of choosing w_{nm} can be found in [1]. Effectively the TFK is a weighted summation kernel as given below:

$$K_{\text{TFK}}(\mathbf{Y}_i, \mathbf{Y}_j) = \sum_{n=1}^{N} \sum_{m=1}^{M} w_{nm} K_{\text{LK}}(\mathbf{y}_{in}, \mathbf{y}_{jm}) \tag{7}$$

An illustration of matching a BOVW vector $\mathbf{y}_{j9}$ from the sequence $\mathbf{Y}_j$ with all the BOVW vectors of $\mathbf{Y}_i$ is given in Fig. 3. In this illustration, length of $\mathbf{Y}_i$ is 11 and that of $\mathbf{Y}_j$ is 17. For every match between $\mathbf{y}_{j9}$ with the BOVW vectors from $\mathbf{Y}_i$, a weight $w_{9i}, i = 1, 2, ...11$ is considered. This is because $\mathbf{y}_{j9}$ and $\mathbf{y}_{i6}$ correspond to the center of activity videos where the match needs to be maximum. In (10), a linear kernel is used for matching $\mathbf{y}_{in}$ and $\mathbf{y}_{jm}$. In principle any kernels on fixed-length representation of examples can be used in place of $K_{\text{LK}}(\mathbf{y}_{in}, \mathbf{y}_{jm})$. It is also helpful if better ways of matching a pair of BOVW vectors can be explored. In the nect section we present modified time flexible kernel (MTFK).

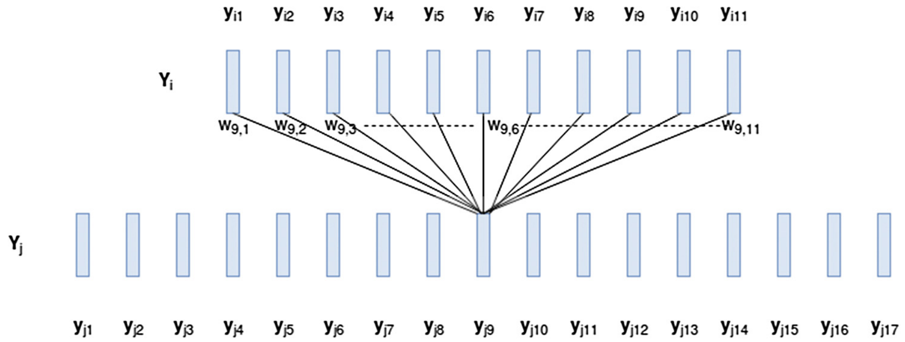

Fig. 3. An illustration of matching $\mathbf{y}_{j9}$ from $\mathbf{Y}_j$ with all the BOVW vectors from $\mathbf{Y}_i$ using suitable weights. Here, the length of $\mathbf{Y}_i$ is 11 and that of $\mathbf{Y}_j$ is 17.

3.2 Modified Time Flexible Kernel

In the design of modified time flexible kernel, we propose to explore better ways of matching a pair of BOVW vectors. The widely used non linear kernels on fixed-length representations such as the Gaussian kernel (GK) or the polynomial kernel (PK) can be used. However, the computation of these kernels require fine tuning of kernel parameters which is not easy. In this work, each video segment is represented using BOVW representation which is a histogram vector representation. For such representations, the frequency-based kernels for SVMs are found to be more effective [6]. So, we have explored two frequency based kernels namely, Histogram Intersection Kernel (HIK) [7] and Hellinger's Kernel (HK) [8] as non linear kernels to modify the TFK.

HIK-Based Modified TFK. Let $\mathbf{y}_{in} = [y_{in1}, y_{in2}, ..., y_{inK}]^T$ and $\mathbf{y}_{jm} = [y_{jm1}, y_{jm2}, ..., y_{jmK}]^T$ be the n^{th} and m^{th} elements of the sequence of BOVW vectors $\mathbf{Y}_i$ and $\mathbf{Y}_j$ corresponding to the video sequences $\mathbf{X}_i$ and $\mathbf{X}_j$ respectively.

The number of matches in the k^{th} bin of the histogram is given by histogram intersection function as,

$$s_k = min(y_{ink}, y_{jmk}) \tag{8}$$

HIK is computed as the total number of matches given by [7],

$$K_{\text{HIK}}(\mathbf{y}_{in}, \mathbf{y}_{jm}) = \sum_{k=1}^{K} s_k \tag{9}$$

HIK-based modified TFK is given by,

$$K_{\text{HIKMTFK}}(\mathbf{Y}_i, \mathbf{Y}_j) = \sum_{n=1}^{N} \sum_{m=1}^{M} w_{nm} K_{\text{HIK}}(\mathbf{y}_{in}, \mathbf{y}_{jm}) \tag{10}$$

HK-Based Modified TFK. In Hellinger's kernel the number of matches in the k^{th} bin of the histogram given by,

$$s_k = \sqrt{y_{ink} y_{jmk}} \tag{11}$$

Hellinger's kernel is computed as the total number of matches across the histogram. It is given by,

$$K_{\text{HK}}(\mathbf{y}_{in}, \mathbf{y}_{jm}) = \sum_{k=1}^{K} s_k \tag{12}$$

HK-based modified TFK is given by:

$$K_{\text{HKMTFK}}(\mathbf{Y}_i, \mathbf{Y}_j) = \sum_{n=1}^{N} \sum_{m=1}^{M} w_{nm} K_{\text{HK}}(\mathbf{y}_{in}, \mathbf{y}_{jm}) \tag{13}$$

When video segment is encoded into BOVW vector, we have proposed HK-based MTFK and HIK-based MTFK. When the encoding scheme is Fisher encoding, we propose improved Fisher kernel (IFK) based modified TFK.

IFK-Based Modified TFK. Let $\mathbf{F}_i = (\mathbf{f}_{i1}, \mathbf{f}_{i2}, ...\mathbf{f}_{in}, ..., \mathbf{f}_{iN})$ and $\mathbf{F}_j = (\mathbf{f}_{j1}, \mathbf{f}_{j2}, ...\mathbf{f}_{jn},\ ..., \mathbf{f}_{jN})$ be the sequence of Fisher score vector representation for i^{th} and j^{th} video. The improved Fisher kernel between $\mathbf{f}_{in}$ and $\mathbf{f}_{jm}$ is computed as,

$$K_{\text{IFK}}(\mathbf{f}_{in}, \mathbf{f}_{jm}) = \mathbf{f}_{in}^T \mathbf{f}_{jm} \tag{14}$$

IFK-based modified TFK is given by,

$$K_{\text{IFMTFK}}(\mathbf{F}_i, \mathbf{F}_j) = \sum_{n=1}^{N} \sum_{m=1}^{M} w_{nm} K_{\text{IFK}}(\mathbf{f}_{in}, \mathbf{f}_{jm}) \tag{15}$$

In the design of TFK and MTFK, a video is represented as a sequence of segments. It is possible for a sequence of sub-activities occurring one after the other to form a higher level of sub-activity. For example, in the video of cricket game, sub-activities such as 'running' and 'throwing a ball' form a higher level sub-activity, 'bowling'. Hence, it may be desirable to match a pair of videos at different levels of sub-activities. For this, we present segment level pyramid match kernel in the next section.

Combining Kernels. To take the maximum advantage of the video representation, we consider the BOVW encoding of the entire video $\mathbf{z}_i$ and the sequence of BOVW representation, $\mathbf{Y}_i$ of a video sequence, $\mathbf{X}_i$. We consider linear combination of kernels, MTFK and kernel computed on BOVW encoding of entire video.

$$K_{COMB}(\mathbf{X}_i, \mathbf{X}_j) = K_1(\mathbf{Y}_i, \mathbf{Y}_j) + K_2(\mathbf{z}_i, \mathbf{z}_j) \tag{16}$$

Here, $K_1(\mathbf{Y}_i, \mathbf{Y}_j)$ is either HIK-based MTFK or HK-based MTFK and $K_2(\mathbf{z}_i, \mathbf{z}_j)$ is LK or HIK or HK.

The base kernel (LK or HIK or HK) is a valid positive semidefinite kernel and multiplying a valid positive semidefinite kernel by a scalar is a valid positive semidefinite kernel [15]. Also, the sum of valid positive semidefinite kernels is a valid positive semidefinite kernel [15]. Hence, both TFK and modified TFK are valid positive semidefinite kernels.

3.3 Segment Level Pyramid Match Kernel

An activity video is a sequence of sub-activities. To design segment level pyramid match kernel (SLPMK), a video is decomposed into increasingly finer segments and is represented as a pyramid of segments. To compute segment level pyramid match kernel (SLPMK) between two videos, we match the corresponding video segments at each level of the pyramid. Let $l = 0, 1, ..., L-1$ be the L levels of the pyramid [32]. At 0^{th} level, complete video sequence is considered as a segment. At the 1^{st} level, video sequence is divided into two equal segments. At the 2^{nd} level, a video sequence is divided into four equal segments and so on. Hence, at any level l, a video sequence is divided into 2^l equal segments. Every segment is encoded into a BOVW vector before matching. In this work, we propose to use two approaches to encode a video segment into a BOVW vector. In the first approach codebook based encoding (CBE) is used. In the second approach, GMM-based encoding (GMMe) is used. The design of SLPMK when CBE and GMME are used is presented as follows.

Codebook Based SLPMK. Let $\mathbf{X}_i$ and $\mathbf{X}_j$ be the i^{th} and j^{th} videos to be matched using the proposed SLPMK. At the l^{th} level of the pyramid, let $\mathbf{y}_{lp}$ be the K-dimensional BOVW vector corresponding to p^{th} segment. Let y_{lpk} be the k^{th} element of $\mathbf{y}_{lp}$ that corresponds to the number of feature vectors of p^{th} segment assigned to the k^{th} codeword. The number of matches in the k^{th} codeword between the p^{th} segments of $\mathbf{X}_i$ and $\mathbf{X}_j$ at l^{th} level of pyramid is given by,

$$s_{lpk} = min(y_{ilpk}, y_{jlpk}) \tag{17}$$

Total number of matches at level l between the p^{th} segments of $\mathbf{X}_i$ and $\mathbf{X}_j$ is given by,

$$S_{lp} = \sum_{k=1}^{K} s_{lpk} \tag{18}$$

Total number of matches between $\mathbf{X}_i$ and $\mathbf{X}_j$ at level l is obtained as,

$$\hat{S}_l = \sum_{p=1}^{2^l} S_{lp} \tag{19}$$

The number of matches found at level l also includes all the matches found at the finer level $l+1$. Therefore, the number of new matches found at level l is given by $\hat{S}_l - \hat{S}_{l+1}$. The codebook based segment level pyramid match kernel (CBSLPMK) is computed as:

$$K_{\text{CBSLPMK}}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{l=0}^{L-2} \frac{1}{2^{L-(l+1)}} (\hat{S}_l - \hat{S}_{l+1}) + \hat{S}_{L-1} \tag{20}$$

In this method, K-means clustering is used to construct bag-of-codewords. Soft clustering could be used to construct a better SLPMK. In the next subsection, we use GMM-based SLPMK. Information about the spread, the size of clusters along with the centers of clusters is considered in GMM- based soft assignment of feature vectors.

GMM-Based SLPMK. Here, a segment of a video is encoded using GMME. This involves building a GMM using all the feature vectors of all the sequences of all the activity classes. The soft assignment of a feature vector $\mathbf{x}_t$ of a segment of a video to the k^{th} component of GMM is given by the responsibility term,

$$\gamma_k(\mathbf{x}_t) = \frac{w_k \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_k, \boldsymbol{C}_k)}{\sum_{j=1}^{K} w_j \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_j, \boldsymbol{C}_j)} \tag{21}$$

where, $\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{C}_k)$ is the k^{th} Gaussian component with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{C}_k$. Here, w_k denotes the mixture weight. For the p^{th} video segment at l^{th} level of pyramid, the effective number of feature vectors of a sequence $\mathbf{X}$ assigned to the component k given by,

$$y_{lpk} = \sum_{t=1}^{T} \gamma_k(\mathbf{x}_t) \tag{22}$$

where, T is the number of feature vectors in the p^{th} segment of $\mathbf{X}$. For a pair of examples represented as sequence of feature vectors, $\mathbf{X}_i$ and $\mathbf{X}_j$, number of matches in the k^{th} codeword between the p^{th} segments of $\mathbf{X}_i$ and $\mathbf{X}_j$ at l^{th} level of pyramid (s_{lpk}), total number of matches at level l between the p^{th} segments (s_{lp}) and total number of matches between $\mathbf{X}_i$ and $\mathbf{X}_j$ at level l ($\hat{S}_l$) are computed as in (17), (18) and (19) respectively. GMMSLPMK between a pair of videos $\mathbf{X}_i$ and $\mathbf{X}_j$, K_{GMMSLPMK} is then computed as in (20). In the next section, we present segment level probabilistic sequence kernel.

3.4 Segment Level Probabilistic Sequence Kernel

Segment level probabilistic sequence kernel (SLPSK) [32] divides a video sequence into a fixed number of segments and then maps each segment onto a probabilistic feature vector. SLPSK between a pair of videos is obtained by matching the corresponding segments. Let $\mathbf{X}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, ...\mathbf{x}_{it}, ...\mathbf{x}_{iT_i})$ and $\mathbf{X}_j = (\mathbf{x}_{j1}, \mathbf{x}_{j2}, ...\mathbf{x}_{jt}, ...\mathbf{x}_{jT_j})$ be the sequence of feature vectors representation corresponding to i^{th} and j^{th} video. Let $\mathbf{X}_i$ and $\mathbf{X}_j$ be divided into N segments, such that $\mathbf{X}_i = (\mathbf{X}_i^1, \mathbf{X}_i^2, ...\mathbf{X}_i^n, ...\mathbf{X}_i^N)$ and $\mathbf{X}_j = (\mathbf{X}_j^1, \mathbf{X}_j^2, ...\mathbf{X}_j^n, ...\mathbf{X}_j^N)$ be the sequence of segments corresponding to $\mathbf{X}_i$ and $\mathbf{X}_j$ respectively. Here, let $\mathbf{X}_i^n = (\mathbf{x}_{i1}^n, \mathbf{x}_{i2}^n, ...\mathbf{x}_{it}^n, ...\mathbf{x}_{iT_{in}}^n)$ be the sequence of feature vectors corresponding to the n^{th} segment in the i^{th} video. In the design of SLPSK, probabilistic sequence kernel (PSK) is computed between corresponding segments of the two videos.

The PSK uses universal background model (UBM) with K components and the class-specific GMMs obtained by adapting to the UBM. The UBM, also called as class independent GMM (CIGMM), is a large GMM built using the training data of all the classes. A feature vector $\mathbf{x}_{it}^n$ corresponding to the t^{th} feature vector in the n^{th} segment of i^{th} video, is represented in a higher dimensional feature space as a vector of responsibility terms of the $2K$ components (K from class-specific adapted GMM and other K from UBM). $\Psi(\mathbf{x}_{it}^n) = [\gamma_1(\mathbf{x}_{it}^n), \gamma_2(\mathbf{x}_{it}^n), ..., \gamma_{2K}(\mathbf{x}_{it}^n)]^T$. Since the element $\gamma_k(\mathbf{x}_{it}^n)$ indicates the probabilistic alignment of $\mathbf{x}_{it}^n$ to the k^{th} component, $\Psi(\mathbf{x}_{it}^n)$ is called the probabilistic alignment vector which includes the information common to all the classes. A sequence of feature vectors $\mathbf{X}_i^n$ corresponding to the n^{th} segment of the i^{th} video is represented as a fixed dimensional vector $\Phi_{\text{PSK}}^n(\mathbf{X}_i^n)$, and is given by,

$$\Phi_{\text{PSK}}^n(\mathbf{X}_i^n) = \frac{1}{T_{in}} \sum_{i=1}^{T_{in}} \Psi(\mathbf{x}_{it}^n) \tag{23}$$

Then, the PSK between two segments $\mathbf{X}_i^n$ and $\mathbf{X}_j^n$ is computed as,

$$K_{\text{PSK}}^n(\mathbf{X}_i^n, \mathbf{X}_j^n) = \Phi_{\text{PSK}}^n(\mathbf{X}_i^n)^T S_n^{-1} \Phi_{\text{PSK}}^n(\mathbf{X}_j^n) \tag{24}$$

where, S_n is the correlation matrix given by,

$$S_n = \frac{1}{M_n} R_n^T R_n \tag{25}$$

where, R_n is the matrix whose rows are the probabilistic alignment vectors for feature vectors of the n^{th} segment and M_n is the total number of feature vectors in the n^{th} segment.

The SLPSK between $\mathbf{X}_i$ and $\mathbf{X}_j$ is then computed as combination of the segment-specific PSKs as follows,

$$K_{\text{SLPSK}}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{n=1}^{N} K_{\text{PSK}}^n(\mathbf{X}_i^n, \mathbf{X}_j^n) \tag{26}$$

Since, PSK is a valid semidefinite kernel, the segment specific PSK is also a valid positive semidefininte kernel. Hence, SLPSK is also a valid positive semidefinite kernel because the sum of valid positive semidefinite kernel is a valid positive semidefinite kernel. In the next section, we present segment level Fisher kernel.

3.5 Segment Level Fisher Kernel

To design segment level Fisher kernel (SLFK), a video represented as a sequence of feature vectors is divided into a fixed number of segments. Computation of SLFK involves computing Fisher kernel (FK) between the corresponding segments of the two video sequences and then combining the segment level match to get SLFK between the two videos. Computation of FK between a pair of segments involves first encoding the segments into respective Fisher score vectors [23]. Let $\mathbf{X}_i^n = (\mathbf{x}_{i1}^n, \mathbf{x}_{i2}^n, ...\mathbf{x}_{it}^n, ...\mathbf{x}_{iT_{in}}^n)$ and $\mathbf{X}_j^n = (\mathbf{x}_{j1}^n, \mathbf{x}_{j2}^n, ...\mathbf{x}_{jt}^n, ...\mathbf{x}_{jT_{jm}}^n)$ be the n^{th} segment of the i^{th} and j^{th} videos. Let $\mathbf{X}_i^n$ and $\mathbf{X}_j^n$ be encoded into the Fisher score vectors $\mathbf{f}_i^n$ and $\mathbf{f}_j^n$ respectively.

The improved Fisher kernel (IFK) between two segments of the videos $\mathbf{X}_i^n$ and $\mathbf{X}_j^n$ is computed as,

$$K_{\text{FK}}^n(\mathbf{X}_i^n, \mathbf{X}_j^n) = (\mathbf{f}_i^n)^T(\mathbf{f}_j^n)^T \tag{27}$$

The SLFK for $\mathbf{X}_i$ and $\mathbf{X}_j$ is then computed as combination of the segment-specific FKs as follows,

$$K_{\text{SLFK}}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{n=1}^{N} K_{\text{FK}}^n(\mathbf{X}_i^n, \mathbf{X}_j^n) \tag{28}$$

Since, FK is a valid semidefinite kernel, the segment specific FK is also a valid positive semidefininte kernel. Hence, SLFK is also a valid positive semidef- inite kernel because the sum of valid positive semidefinite kernel is a valid positive semidefinite kernel. In the next section, we present combination of kernels.

4 Experimental Studies

In this section, we present the results of experimental studies carried out to verify the effectiveness of sequence kernels for video activity recognition. We first present the datasets used and video representation considered. Then, we present the various studies conducted.

4.1 Datasets

UCF Sports Dataset. This dataset comprises a collection of 150 sports videos of 10 activity classes. On an average, each class contains about 15 videos with an average length of each video being approximately 6.4 s. We follow leave-one-out cross validation strategy as used in [35].

UCF50 Dataset. UCF50 is an action recognition data set with 50 action categories, consisting of realistic videos taken from youtube. This data set is an extension of YouTube Action data set. This dataset contains 6681 video clips of 50 different activities. We follow leave-one-group-out cross-validation strategy used in [34].

Hollywood 2 Dataset. This dataset contains 1707 video clips belonging to 12 classes of human actions. It has 823 videos in the training dataset and 884 videos in the test dataset as used in [31].

4.2 Video Representation

Each video is represented using improved dense trajectories (IDT) descriptor [10]. IDT descriptor densely samples feature points in each frame and tracks them in the video based on optical flow. To incorporate the temporal information, IDT descriptor is extracted using a sliding window of 30 frames with an overlap of 15 frames. For a particular sliding window, multiple IDT descriptors, each of 426 dimensions are extracted. The 426 features of an IDT descriptor comprise multiple descriptors such as histogram of oriented gradient (HOG), histogram of optical flow (HOF), and motion boundary histograms (MBH). The number of descriptors per window depends on the number of feature points tracked in that window. A video clip is represented as a sequence of IDT descriptors. We propose to encode sequence of feature vectors of a video in two ways. In the first approach, entire video is encoded into a bag-of-visual-words (BOVW) representation. In the second approach, a video is encoded into a sequence of BOVW representation. In this work, we propose to consider the GMM-based soft clustering approach for video encoding. We also compare the GMM-based encoding (GMME) approach with the codebook-based encoding (CBE) approach proposed in [1]. In this work, we study the effectiveness of different sequence kernels for activity recognition using SVM-based classifiers.

4.3 Studies on Activity Recognition in Video Using BOVW Representation Corresponding to Entire Videos

In this section, we study the SVM-based activity recognition by encoding an entire video clip into a single BOVW vector representation. We consider codebook-based encoding (CBE) [1] and GMM-based encoding (GMME) methods for generating BOVW representation. Different values for the codebook size in CBE and the number of clusters in GMM, is explored and an optimal value of 256 is chosen. For SVM-based classifier, we need a suitable kernel. We propose to consider linear kernel (LK), histogram intersection kernel (HIK) and Hellinger's kernel (HK). The accuracy of activity recognition in videos using SVM-based classifier for the three datasets is given in Table 1. It is seen from Table 1 that video activity recognition using SVM-based classifier that uses the frequency based kernels, HIK and HK, is better than that using LK. This shows the suitability of frequency based kernels when the videos are encoded into BOVW representation. It is also seen that the performance of SVM-based classifier using GMME is better than that using CBE used in [1]. This shows the effectiveness of GMM-based soft clustering approach for video encoding. In the next section, we study SVM-based approach to video activity recognition when a video clip is represented as a sequence of BOVW representation.

Table 1. Accuracy in (%) of SVM-based classifier for activity recognition in videos using linear kernel (LK), Hellinger's kernel (HK) and histogram intersection kernel (HIK) on the BOVW encoding of the entire video. Here CBE corresponds to code book based encoding proposed in [1] and GMME corresponds to GMM-based video encoding proposed in this work.

	Datasets					
	UCF Sports		UCF50		Hollywood	
Kernel/Encoding	CBE	GMME	CBE	GMME	CBE	GMME
LK	80.67	90.40	80.40	90.40	80.40	90.40
HK	83.33	92.93	83.33	92.27	83.00	93.57
HIK	84.67	92.53	82.27	92.53	83.00	93.13

4.4 Studies on Activity Recognition in Videos Using Sequence of Feature Vectors Representation for Videos

In this section, we study the activity recognition in videos using sequence of feature vectors representation of videos. To build an SVM-based activity recognizer, we propose to consider time flexible kernel (TFK), modified time flexible kernel (MTFK), segment level pyramid match kernel (SLPMK), segment level probabilistic sequence kernel (SLPSK) and segment level Fisher kernel (SLFK). For computing TFK, a sequence of feature vectors is encoded into a sequence of BOVW vectors where a segment of 30 frames is considered. We consider both codebook based encoding (CBE) and GMM-based encoding (GMME) approaches. Different values for the codebook size in CBE and the number of clusters in GMM, is explored and an optimal value of 256 is chosen. MTFK between a pair of videos is computed using the same sequence of BOVW vectors representation used in TFK. Here, we consider the frequency based kernels HK and HIK for MTFK. The accuracy of SVM-based activity recognizer for the three datasets using the proposed kernels is given in Table 2. It is seen from Table 2 that, MTFK-based SVMs give better performance than TFK-based SVMs. This shows the effectiveness of the kernels proposed in this work. From Tables 1 and 2, it is also seen that the TFK-based SVMs give better performance than LK-based SVMs that use BOVW vector representation of entire video. This shows the importance of using the temporal information of video for the activity recognition. It is also seen that the performance of MTFK-based SVMs is better than the SVM-based classifiers using the frequency based kernels, HK and HIK, on BOVW vectors encoding corresponding to the entire video.

It is seen from Table 2 that the performance of MTFK-based activity recognizer is comparable to that obtained using the activity recognizers that use SLPMK, SLPSK and SLFKs for UCF Sports and UCF50 datasets. In case of Hollywood dataset, SLFK based SVM classifier is found to perform better than the other classifiers. For SLPMK we have considered levels $L = 1, 2, 3$ and at each level a segment of video is encoded using CBE and GMME. Different values for the codebook size in CBE and the number of clusters in GMM, is explored and an optimal value of 256 is chosen. For SLPSK and SLFK, we divide the video into segments in 3 ways. The first approach involves dividing a video into two segments. In the second approach, a video is divided into 4 segments and in the third approach there are 8 segments considered.

Table 2. Accuracy in (%) of SVM-based classifier for activity recognition in videos using TFK, MTFK, SLPMK, SLPSK and SLFK computed on sequence of BOVW vectors representation of videos. Here, CBE corresponds to code book based encoding proposed in [1] and GMME corresponds to GMM-based video encoding proposed in this work. HK-MTFK corresponds to HK-based modified TFK and HIK-MTFK denotes the HIK-based modified TFK. SLPMK with three levels $L = 1, 2, 3$ is considered. Here SLPSK-1 denotes SLPSK computed by diving a video into two segments. SLPSK-2 and SLPSK-3 correspond to SLPSK computed by diving a video into 4 and 8 segments respectively. SLFK-1, SLFK-2 and SLFK-3 correspond to SLFK computed by diving a video into 2, 4 and 8 segments respectively.

	Datasets					
	UCF Sports		UCF50		Hollywood	
Kernel/Encoding	CBE	GMME	CBE	GMME	CBE	GMME
TFK	82.00	91.27	81.67	91.27	82.67	91.67
HK-MTFK	86.67	95.73	86.67	95.27	86.13	92.26
HIK-MTFK	86.00	95.60	86.00	95.00	86.00	92.00
SLPMK (L = 1)	84.50	95.60	85.07	94.06	85.00	94.00
SLPMK (L = 2)	85.00	96.00	85.60	94.50	85.60	94.50
SLPMK (L = 3)	84.50	96.00	85.67	95.00	83.00	94.07
FK-MTFK	95.67		95.00		95.07	
SLPSK-1	92.67		93.00		92.67	
SLPSK-2	93.67		94.00		94.00	
SLPSK-3	93.27		93.70		93.00	
SLFK-1	95.00		94.60		94.00	
SLFK-2	95.27		94.67		95.00	
SLFK-3	95.07		94.67		95.00	

4.5 Studies on Activity Recognition in Video Using Combination of Kernels

In this section, we combine a kernel computed on BOVW representation of entire videos and a kernel computed on sequence of BOVW vectors' representation of video. We consider simple additive combination so that a combined kernel $COMB(K_1 + K_2)$ corresponds to addition of K_1 and K_2. Here K_1 corresponds to kernel computed using sequence of BOVW vectors representation of videos, and K_2 corresponds to the kernel computed on the BOVW representation of entire video. The performance of SVM-based classifier using combined kernels for video activity recognition is given in Table 3. It is seen from Table 3 that the accuracy of SVM-based classifier using the combination kernel involving TFK is better than that for the SVM-based classifier using only TFK. This shows the effectiveness of the combination of kernels. It is also seen that the performance for SVM-based classifier using combination of kernels involving HK and HIK computed on entire video is better than that obtained with combination of kernel involving LK computed on entire video. It is also seen that the performance of SVM-based classifiers using combination kernel involving MTFKs is better than that using TFKs. This shows the effectiveness of the proposed MTFK in activity recognition in videos. In Figs. 4, 5 and 6 we compare the performance of SVM-based activity recognition using different kernels for the UCF sports dataset, UCF 50 dataset and Hollywood datasets respectively. It is seen from Figs. 4, 5 and 6 that, the SVM-based activity recognizers using the combination of kernels performs better than classifiers that use other kernels.

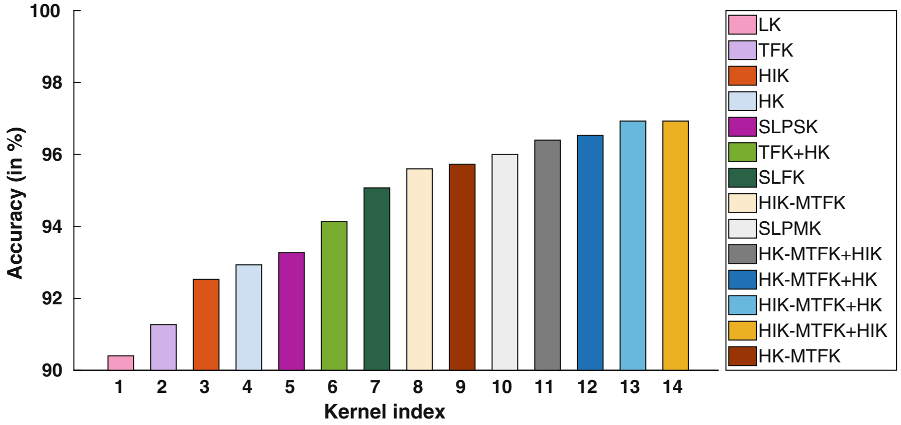

Fig. 4. Comparison of performance of SVM-based classifier using different kernels for the video activity recognition in UCF Sports dataset.

Table 3. Comparison of performance of activity recognition in video using SVM-based classifier that uses combination of kernels. Here, $COMB(K_1 + K_2)$ indicate additive combination of kernels K_1 and K_2 respectively. K_1 is a kernel computed on the sequence of BOVW vectors representation of videos and K_2 is a kernel computed on the BOVW representation of the entire video. Here, CBE corresponds to code book based encoding proposed in [1] and GMME corresponds to GMM-based video encoding proposed in this work.

	Datasets					
	UCF Sports		UCF50		Hollywood	
Kernel/Encoding	CBE	GMME	CBE	GMME	CBE	GMME
COMB(TFK+LK)	82.67	93.87	82	93.87	82.67	93.87
COMB(HK-MTFK+LK)	84.67	94.13	84.67	93.87	83.13	95
COMB(HIK-MTFK+LK)	82.67	94.13	83.67	93.87	83	95.13
COMB(TFK+HK)	86	94.13	85.13	94.13	85.13	94.26
COMB(HK-MTFK+HK)	86.67	96.53	86	96.13	86.26	96.67
COMB(HIK-MTFK+HK)	87.33	96.93	86.93	96.27	86	96.13
COMB(TFK+HIK)	84	94.67	84	94.67	84.26	94.13
COMB(HK-MTFK+HIK)	86.67	96.4	86.67	96.4	86.67	96
COMB(HIK-MTFK+HIK)	86	96.93	86.27	96.27	86	96.27

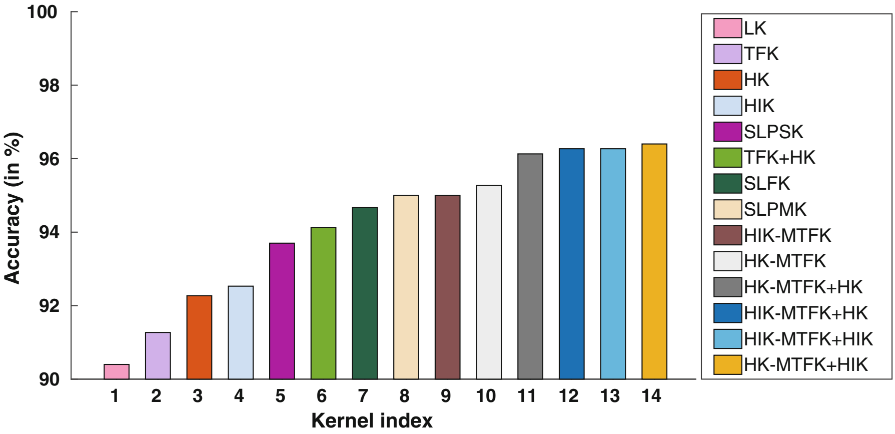

Fig. 5. Comparison of performance of SVM-based classifier using different kernels for the video activity recognition in UCF50 dataset.

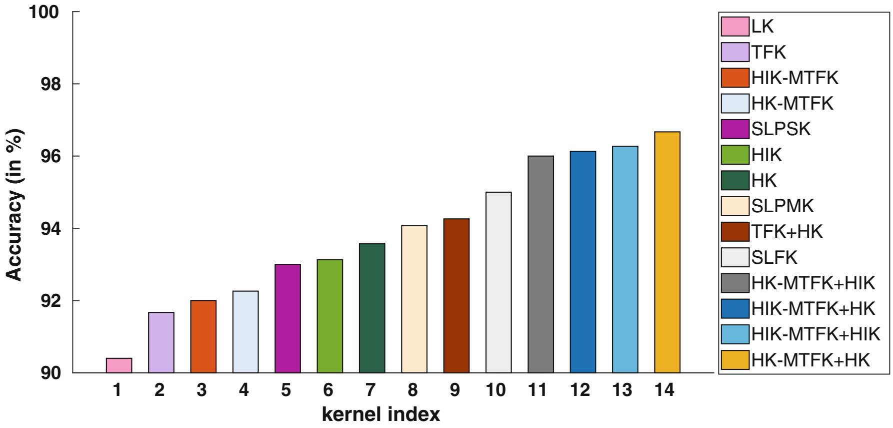

Fig. 6. Comparison of performance of SVM-based classifier using different kernels for the video activity recognition in hollywood dataset.

5 Conclusion

In this paper, we proposed approaches to SVM-based video activity recognition using different sequence kernels. A video activity comprises of a sequence of sub-activities whose time ordering is important for discriminating one video activity from the other. A sub-activity corresponds to a small segment of the video. Hence, in this work, we proposed sequence kernels that consider video segments. In this paper, we proposed modified time flexible kernel, segment level pyramid match kernel, segment level probabilistic sequence kernel and segment level Fisher kernel. The studies conducted using bench mark datasets show the effectiveness of the proposed kernels for SVM based activity recognition.

References

1. Rodriguez, M., Orrite, C., Medrano, C., Makris, D.: A time flexible kernel framework for video-based activity recognition. Image Vis. Comput. **48**, 26–36 (2016)
2. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 379–385 (1992)
3. Shabou, A., LeBorgne, H.: Locality-constrained and spatially regularized coding for scene categorization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3618–3625 (2012)
4. Wang, J., Liu, P., She, M., Liu, H.: Human action categorization using conditional random field. In: IEEE Workshop on Robotic Intelligence in Informationally Structured Space (RiiSS), pp. 131–135 (2011)
5. Dileep, A.D., Sekhar, C.C.: HMM based intermediate matching kernel for classification of sequential patterns of speech using support vector machines. IEEE Trans. Audio Speech Lang. Process. **21**(12), 2570–2582 (2013)

6. Sharma, N., Sharma, A., Thenkanidiyoor, V., Dileep, A.D.: Text classification using combined sparse representation classifiers and support vector machines. In: 4th International Symposium on Computational and Business Intelligence (ISCBI), pp. 181–185 (2016)
7. Van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., Geusebroek, J.-M.: Visual word ambiguity. IEEE Trans. Pattern Anal. Mach. Intell. **32**(17), 1271–1283 (2010)
8. Chatfield, K., Lempitsky, V.S., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. BMVC **2**(4), 8 (2011)
9. Soomro, K., Zamir, A.R.: Action recognition in realistic sports videos. In: Moeslund, T.B., Thomas, G., Hilton, A. (eds.) Computer Vision in Sports. ACVPR, pp. 181–208. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09396-3_9
10. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3551–3558 (2013)
11. Xu, D., Chang, S.-F.: Video event recognition using kernel methods with multilevel temporal alignment. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1985–1997 (2008)
12. Cao, L., Mu, Y., Natsev, A., Chang, S.-F., Hua, G., Smith, J.R.: Scene aligned pooling for complex video recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, pp. 688–701. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_49
13. Vahdat, A., Cannons, K., Mori, G., Oh, S., Kim, I.: Compositional models for video event detection: a multiple kernel learning latent variable approach. In: IEEE International Conference on Computer Vision (ICCV), pp. 1185–1192 (2013)
14. Li, W., Yu, Q., Divakaran, A., Vasconcelos, N.: Dynamic pooling for complex event recognition. In: IEEE International Conference on Computer Vision (ICCV), pp. 2728–2735 (2013)
15. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
16. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 357–360 (2007)
17. Klaser, A., Marszałek, M., Schmid, C.: A Spatio-temporal descriptor based on 3D-gradients. In: 19th British Machine Vision Conference (BMVC), pp. 1–275 (2008)
18. Laptev, I.: Space-time interest points. Int. J. Comput. Vis. **64**(2–3), 107–123 (2005)
19. Kliper-Gross, O., Gurovich, Y., Hassner, T., Wolf, L.: Motion interchange patterns for action recognition in unconstrained videos. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577, pp. 256–269. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_19
20. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), New York, vol. 2, pp. 2169–2178 (2006)
21. Gupta, S., Dileep, A.D., Thenkanidiyoor, V.: Segment-level pyramid match kernels for the classification of varying length patterns of speech using SVMs. In: 24th European Signal Processing Conference (EUSIPCO), pp. 2030–2034 (2016)
22. Zha, S., Luisier, F., Andrews, W., Srivastava, N., Salakhutdinov, R.: Exploiting image-trained CNN architectures for unconstrained video classification. arXiv preprint arXiv:1503.04144 (2015)

23. Wu, Z., Wang, X., Jiang, Y.-G., Ye, H., Xue, X.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification, pp. pp. 461–470. arXiv preprint arXiv:1504.01561 (2015)
24. Varadarajan, B., Toderici, G., Vijayanarasimhan, S., Natsev, A.: Efficient large scale video classification. arXiv preprint arXiv:1505.06250 (2015)
25. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 1725–1732 (2014)
26. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
27. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4305–4314 (2015)
28. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream ConvNets. arXiv preprint arXiv:1507.02159 (2015)
29. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409–1556 (2014)
31. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2929–2936 (2009)
32. Gupta, S., Thenkanidiyoor, V., Aroor Dinesh, D.: Segment-level probabilistic sequence kernel based support vector machines for classification of varying length patterns of speech. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) ICONIP 2016. LNCS, vol. 9950, pp. 321–328. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46681-1_39
33. Thenkanidiyoor, V., Chandra Sekhar, C.: Dynamic kernels based approaches to analysis of varying length patterns in speech and image processing tasks. In: Pattern Recognition And Big Data. World Scientific (2017)
34. Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. Mach. Vis. Appl. J. (MVAP) **24**, 971–981 (2012)
35. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
36. Sharma, A., Kumar, A., Allappa, S., Thenkanidiyoor, V., Dileep, A.D.: Modified time flexible kernel for video activity recognition using support vector machines. In: 7th International Conference on Pattern Recognition Applications and Methods, pp. 133–140 (2018)