**Computer Communication & Collaboration (Vol. 2, Issue 3, 2014)**
**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

# Highlighting Important Video Segments for Human Action Recognition

***Neda Azouji***

Computer Vision and Pattern Recognition Lab

School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

E-mail: nazouji@cse.shirazu.ac.ir

***Zohreh Azimifar*** (Correspondence author)

Computer Vision and Pattern Recognition Lab

School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

E-mail: azimifar@cse.shirazu.ac.ir

**Abstract:** Human Action Recognition in video has become a very important topic in many computer vision applications such as automated surveillance and human-computer interaction. Various approaches have been developed, but most of them focus on entire video sequence. In this study, we introduce a step to standard action recognition pipeline called "video highlighting". The main contribution of this paper is to propose a new approach, which highlights parts of input video data for recognition task. More informative data can be retained while less important data are eliminated when necessary. First, key-frames are extracted automatically and then, to preserve general dynamics of video sequences, continuous segments are constructed by including frames around the selected key-frames. Finally, a new input video is generated by concatenating these excerpts. This new sequence is used as input for the action recognition framework. The experiments were conducted for two different conventional action datasets: KTH, UCF sports; and the results are reported for various configurations. We give a performance comparison between our highlighted input videos and original video sequences. Furthermore, different low-level features that are used in the key-frame extraction process are tested with different numbers of extracted key-frames. The experiments indicate that the proposed method not only achieves higher performance, but also can maintain and to some extent increase recognition precision.

**Keywords:** Human Action Recognition; Key-frame extraction; Unsupervised feature learning; Bag-of-Words (BoW); Spatio-temporal features; Video sequence

## 1. Introduction

Every day thousands of hours of video are captured across the world by Closed-Circuit Television (CCTV) cameras, webcams, and traffic-cams. Human Action Recognition (HAR) is a hot topic in computer vision research area in which human action patterns are recognized. Humans can easily identify the type of actions performed in a video. However, the automatic recognition of human actions [1] is a challenging task in computer vision. Some of the growing applications of the computer-based HAR are in areas such as automated surveillance [2], content-based video retrieval

[3], video summarization [4], cognitive science [5], and human-computer interaction [3]. Generally, the vision-based human action recognition process has two stages: feature extraction and classification.

In a video processing task, there are two possible situations according to the video data at hand; one is having a large dataset for which analyzing data is a daunting task, the other is having a small dataset for which we are not able to acquire sufficient information. A video sequence normally contains a large number of frames. Generally a frame rate of at least 25fps is required to ensure that humans do not realize any discontinuity in a video stream [6]. Therefore, there are numerous images in a one-hour video clip. This large volume of video data limits many practical applications and consequently, we need methods which reduce the video size while keeping useful information intact. On the other hand, generally in available realistic video data, there are few instances which can be regarded as uncontrolled video data. Few instances are another issue to deal with video sequences. Such realistic data must be captured under real world conditions which cannot be satisfied in artificial experiments; indeed, these video clips should be recorded as a real event occurs and we do not have any control on such events. For instance, this is the case for TV and cinema style movie data, sports broadcasts, music videos, or personal amateur clips. The main challenge in learning from this kind of data is that quite few data are available for the recognition task. Therefore, we should exploit the available data optimally.

A large body of work exists that addresses the topic of automatic human action recognition [1, 7]. Those works achieved varying degrees of success, however most of them focus on analysis of the entire video clip. Due to inherent spatio-temporal redundancies in video data, those approaches are using more information than required. In most videos many periods of time contain little to no activities or events. In other words, events only occur in a small part of the spatio-temporal dimension of a video clip.

In this work, we aim to introduce an efficient method for highlighting important and discriminative parts of video data and removing those segments of video sequences that contain redundant or unnecessary information. Our approach automatically generates an intelligently composed version of a short or long video sequence, which preserves the general dynamics of the original video. The proposed algorithm has two stages: first, we generate new input data by manipulating original data, and then we use this video data as input to a Bag-of-Words (BoW) framework in order to recognize human actions. Our experimental results indicate that the re-generated video sequence is sufficient to achieve a performance competitive to the Original video sequence when used as an input.

The rest of the paper is organized as follows: in Section 2 related research in the field of Human action recognition (HAR) is reviewed. Section 3 provides the details of our proposed framework. In Section 4 the experimental results and discussions are presented. Finally, Section 5 concludes the paper and provides remarks for future studies.

## 2. Related Works

Action recognition and event classification in video data have been studied extensively in recent years. Most of the related methods use either a top-down or a bottom-up approach. A typical top-down approach needs foreground segmentation, which uses a shape or an appearance model of a human for detection or tracking [8]. This method has shown promising results in a constrained environment [1]. But in real-world low resolution videos, in presence of background or camera motion, clutter, or occlusion the segmentation and tracking may not be reliable. Clutters or occlusions act like a hindrance for accurate tracking, while dynamic camera and background make the segmentation process trickier. On the other hand, a typical bottom-up approach [9] can learn a set of visual words by using local spatio-temporal salient features without applying video

segmentation and tracking. This paper focuses on this second method to generalize our algorithm on realistic datasets as well as artificial datasets.

Recently, local image and video features are widely used for many recognition tasks such as object and scene recognition [10, 11] as well as HAR. Local space-time features capture shape and motion characteristics of a video. Moreover, such features provide a representation of events, which is scale and shift invariant in spatio-temporal space. Furthermore, this representation is independent from background clutter and multiple motions in the scene. Since such local features are usually extracted directly from video data, they are not affected by possible failures of pre-processing methods such as motion segmentation and tracking [12]. In recent years, low-level hand-designed features are heavily employed with much success. Common approaches in visual recognition such as SIFT [13] and HOG [14] rely on these features. A weakness of such techniques is that they are domain specific; therefore extending these features to other sensor modalities such as laser scan, text or even video is difficult and time-consuming. Unsupervised feature learning methods such as Sparse Coding [15], Deep Belief Nets [16] and Stacked Autoencoders [17] Have become more common recently in recognition tasks. Those methods learn features directly from data and therefore, have the capability to be generalized.

In a recent work, Wang et al. [12] have combined various low-level features and descriptors. To make a fair comparison, the authors employed the same state-of-the-art processing pipeline as Vector Quantization, feature normalization and $\chi^2$-kernel SVMs. The only change in the pipeline is the use of different feature detection and feature extraction methods. A very interesting consequence of this research is that there is no universal best hand-engineered feature for all datasets; the experiments suggest that using the data in its raw format when learning features may be more advantageous. The authors have showed unsupervised learning is applicable to different domains and also achieves impressive performance in many realistic video datasets. Independent Subspace Analysis (ISA) as unsupervised feature learning is used in their algorithm.

Some results in the cognitive sciences have led to biologically inspired vision systems for action recognition. Jhuang et al. [19] introduced a method which uses a hierarchy of spatio-temporal features with considerable complexity. The authors have extended the static scene recognition model by replacing shape features with motion features. A set of flow filters is used to extract dense local motion information. In this approach input data are processed by units which are sensitive to motion-directions. Then, responded values are pooled locally and are converted to higher-level responses by comparing them to more complex templates. These values are pooled once more and fed into a discriminative classifier.

Schindler and Van Gool [20] proposed a more sophisticated and a more efficient approach. The authors presented a system for action recognition relying on a very short sequence ("Action snippets") of 1-10 frames and evaluated it on standard datasets. The recognition results illustrated that there are very short snippets which are almost as informative as the entire video sequence. For best use of the available information, both shape and motion features were explicitly extracted. Local shapes are extracted within each frame and optical flows are computed between frames. Finally, action classification is done by using multiple one-versus-all SVMs.

A lot of studies have been done works are done in action recognition field and researchers have developed methods, which increase recognition precision as well as efficiency. The most important theme in our study is extracting principle parts of a video sequence, and in this regard, a key-frame-based approach is used. In the HAR field, few approaches are proposed based on key-frames. In terms of frame-based approaches, some researchers adopted the idea of fast forward. A naive uniform sampling on the frames of a video, though simple, may skip fast activities and lose

the important dynamics of a video. Therefore many non-uniform sampling algorithms have been proposed, such as [21-22]. Early attempts at HAR relied on key-frames, used template matching. Carlsson and Sullivan [23] modeled action recognition as a shape matching problem. They considered the recognition of a particular action (forehand and backhand strokes) of a specific person. An action was represented by a single unique pose. In this approach a single key-frame is extracted manually. In fact, the shape is described by edge maps, and recognition is performed by comparing the image shape with the shape in the key-frame. Their experiments showed that this approach can detect all of the forehand actions without any false positives results. This work proved the importance role of shape in human action recognition task, whereas later researches focused on the dynamic aspect of human actions and considered both motion and shape information.

Some researchers select key-frames as the representation of a video. The methods are often known as video summarization. Video summarization techniques attempt to provide a summary of a video by creating smaller videos containing descriptive sections of the original video. Typically, these techniques employ static representations such as key-frames [24], or motion video representations [4, 25]. The key-frames can be selected by using the concepts of objects and object motion trajectory [26]; or exploiting notions from shot segmentation, scene detection/clustering, and domain knowledge, such as the works presented in [24, 27].

Despite conciseness, representing a video by key-frames has a drawback that the dynamics of a video might be lost. In this regard, Hu and Zheng [28] investigated key-frames which are automatically extracted from the image sequence according to the Zernike moment of the silhouette. The authors calculated feature curve of every action in Weizmann dataset and each value on the curve imply to Zernike moment of the silhouette. There are some local maximum and local minimum points on this curve. These points determine motion corners, indicating the person's motion changes. Frames corresponding to these extreme are called action key-frames [28]. Each segment between two minimum points or two maximum points shows one cycle of a repetitive motion. In fact, one action cycle will be extracted from the entire sequence, and then features are extracted only from this cycle.

Several of the summarization algorithms exploit space-time volumes of action. Rodriguez [4] proposed a method in which space-time "worms" are matched with a user-specified query to find actions of interest. Then these worms are condensed by optimizing their temporal shift. In this approach multiple instances of a relevant activity can be displayed simultaneously so that a summarized video may contain multiple actions at the same time. The authors generate a compact video representation of a long sequence automatically, which contains only desired activities while preserving the general dynamics of the original video.

Chang et al. [29] proposed a framework which offers reducing the length of a video according to its content. In the framework, the principal idea is based on the concept of seam removal, proposed by Avidan et al. [30] for resizing an image. By modifying the dynamic programming approach originally employed by Avidan et al. [30], the authors extract smooth 2D sheets and therefore avoid fragmentation. In this work, in order to cope with videos of any lengths, an out-of-core scheme is proposed. The approach could improve performance by two orders of magnitude and offer one order of magnitude memory space reduction, in comparison to the competitive state-of-the-art methods.

## 3. Proposed Method

Our approach to highlighting part of video data for human action recognition (HAR) is composed of two main phases. First, we process the input video by extracting key-frames and selecting some segments instead of the entire sequence. Second, we follow Wang et al.'s [12] experimental setup in their action recognition pipeline, but using our preprocessed video data as input to this framework.

ISSN 2292-1028(Print) 2292-1036(Online)
DOIC: 2292-1036-2014-03-001-32

By doing so, we can easily understand the contribution of highlighting the principle parts of video data. A scheme of our method is illustrated in Figure 1. The methodology steps are elaborated below:

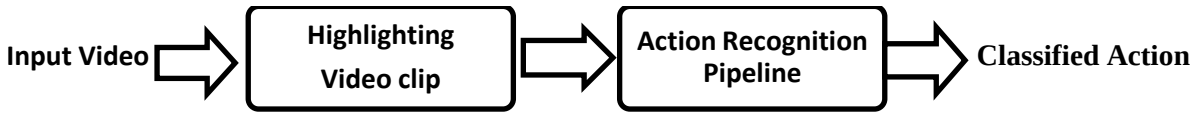


**Figure 1** A scheme of our method. It has two stages: (1) video highlighting, (2) action recognition. We add a step before standard Action Recognition pipeline.

## 3.1 Extracting importance in video data

Theoretically, given a sequence of human actions, all frames in the sequence should be exploited for the recognition task. However, human vision can distinguish common human actions using only few frames. Being inspired by such human capability, we aim to improve our automated action recognition by adding this efficiency. In this section we describe how we identify key-frames in video sequences to preserve the temporal aspect of video data. And we create continuous segments that can be provided as input to the action recognition framework.

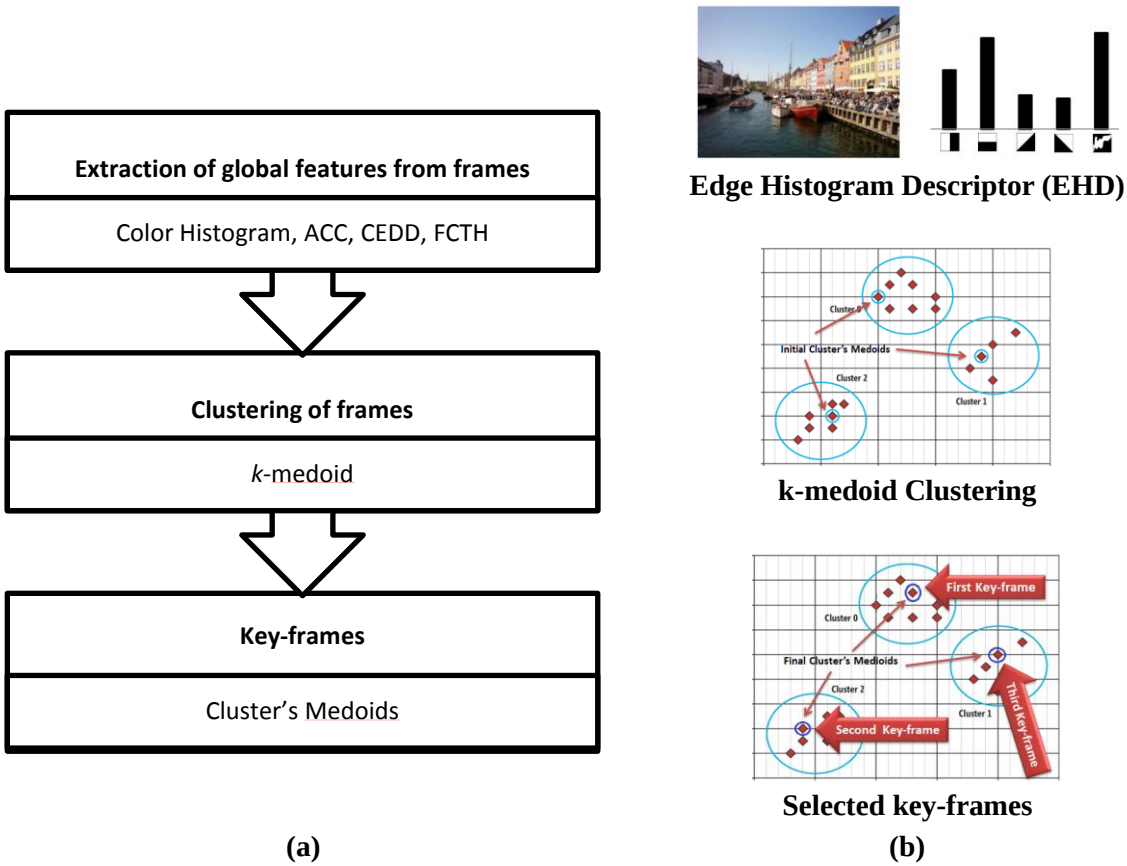


**Figure 2** Process of key-frame extraction. (a) Three main steps in key-frame extraction. (b) An example of each step

### 3.1.1 Key-frame Selection

Key-frames or representative frames are salient images extracted from the underlying video. Therefore, the key-frame set $\mathcal{R}$ is defined as follows [6]:

$$\mathcal{R} = \mathcal{A}_{\text{key-frames}}(\text{V}) = \{\text{f}_{\text{r}_1}, \text{f}_{\text{r}_2}, \dots, \text{f}_{\text{r}_k}\} \quad (1)$$

Where $\mathcal{A}_{\text{key-frames}}$, V, and f denote the key-frame extraction method, video clip, and a frame, respectively.

Since the selection of key-frames from video sequence depends on the information about individual frames, we extract global features of every frame which describes a frame as whole. Most commonly, low-level features such as color histogram or texture features are used for comparing frames. Low-level features are usually applied as the first operation on an image. These features observe every pixel of image to realize if there is a feature present at that pixel. The process of key frame extraction is shown in Figure 2. In this paper, we examine different low-level features and their combination.

**(a)**

**(b)**

**Figure 3** Visualization of three selected key-frames of the "Golf-swing" class in UCF sports dataset. Auto Color Correlogram (ACC) as a low-level feature used in process of key-frame extraction. (a) Sample visualization where the biggest cluster in full size to the left and the other two to the right. (b) Other visualization where key-frames are shown in cluster-size order (medoid of the biggest clusters is the first frame to the left).

Key-frames are selected by using clustering algorithms. All frames in a video clip are assigned into a fixed number of clusters (n). Clustering is performed in the space of low-level features extracted from each frame. All frames of a cluster are visually similar and center of cluster can be considered as a representative key-frame for that cluster. In fact, one image in each cluster can describe the whole set of visually similar images. The actual size of a cluster can specify how much of a video's duration is actually covered by that cluster. Figure 3 demonstrates three selected key-frame of the "Golf-swing" class in UCF sports dataset.

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

The key-frame extraction approach described in this paper consists of the following steps:

a. Extracting global features
b. Clustering frames

### a. Extracting global features

Feature extraction of image is the process of extracting compact, but semantically valuable information from an image. Proper representation of an image necessitates satisfying these two conditions: features should represent contents of the image properly and approaches should efficiently encode the attributes of the image. Generally, an uncompressed video clip is interpreted as a sequence of still images. For each of the images (frames) within a video sequence, we extract certain low-level features.

This section introduces three main features that we propose to use as image representation in first step of key-frame extraction: texture, shape, and color. These are the most common features that can be extracted from an image. Since these features are well-known in video abstraction and each one summarize video sequences from a different point of view, we analyze the impact of using these features in key-frame extraction procedure on action recognition performance. In this paper, we employ nine different combinations of features which are explained in the following subsections.

- **Color Descriptors**

Color is one of the most prominent perceptual features that can be visually recognized by humans and is a powerful image descriptor. Humans regularly use color features to distinguish images. To extract such a feature from the content of an image, a proper color space has to be specified and then an effective color descriptor should be developed. Various color descriptors have been developed based on representation schemes, such as color histograms, color texture, and color correlograms.

**Color Histogram:** Color histogram is the most commonly used technique to represent color feature of an image. Statistically, a color histogram is an approximation of the joint probability of the values of three color channels. Generally, color histogram is obtained by splitting the range of the data into some equally sized bins. After that, for each bin, the number of points from colors of the pixels in an image which fall into each bin are counted and also normalized to total points. It gives the probability of a pixel falling into that bin. Where a color bin is defined as a region of colors, given a color image I(x, y)of size X × Y, which consists of three channels $I = (I_R, I_G, I_B)$, the color histogram used here is

$$h_c(m) = \frac{1}{XY}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}\begin{cases}1 & \text{if } I(x,y) \text{in bin m,}\\ 0 & \text{otherwise}\end{cases} \qquad (2)$$

An issue associated with color histogram is that it considers only the distribution of colors in an image and does not include any spatial information. Several techniques such as Auto Color Correlogram [31] and Color Layout Descriptor [32] are introduced to integrate spatial information with the color histogram.

**Auto Color Correlogram:** A color correlogram [31] represents how the spatial auto-correlation of colors changes with the distance between pixels. Auto Color Correlogram (ACC) considers the

spatial correlation of colors. And it can be used as a spatial extension of the histogram. A color correlogram of an image is a table indexed by color pairs. In such table the kth entry for $(i, j)$ indicates the probability of detecting a pixel with color $j$ at a distance $k$ away from a pixel $i$ in the image [31]. Since this method has high complexity, auto correlogram is used to capture spatial correlation between identical colors only.

We can state color correlogram of an image by answering to the following question: if we pick any pixel $p_1$ of color Ci in the image $I$, and then pick another pixel $p_2$ at distance k away from $p_1$, what is the probability that $p_2$ is also of color $C_i$ [31]?

The auto-correlogram of image I for color $C_i$, with distance $k$ is defined as follows:

$$\gamma_{c_i}^{(k)}(I) \equiv \Pr\left[|p_1 - p_2| = k, (p_2 \in I_{c_i} | p_1 \in I_{c_i})\right] \quad (3)$$

**Color Layout Descriptor:** Another descriptor to capture spatial information is Color Layout Descriptor (CLD) [32]. It is designed to capture spatial distribution of the representative color of a grid superimposed on a region or image. The process of feature extraction includes two parts: grid based dominant color selection and Discrete Cosine Transform (DCT) with quantization. In fact, representation is provided by coefficients of DCT. The CLD is a very compact descriptor which is highly efficient in accelerating browsing and search applications. This descriptor can be applied to still images and video segments.

**Scalable Color Descriptor:** Scalable Color Descriptor (SCD) [33] can be interpreted as a kind of color histogram defined in the Hue-Saturation-Value (HSV) color space with fixed color space quantization. This descriptor uses a Haar transform coefficient encoding. SCD provides scalable representation of an image and also suffers from lack of scalability of feature extraction and matching procedures [33].

- **Shape Descriptors**

Shape is an important visual cue that can be considered as another important low-level image feature. Edge detection and sometimes segmentation are used to extract shape features. Because of lighting effects and occlusion, shape characterization techniques are preferred to local shape segmentation methods. One problem with shape descriptors is that it is hard to find a good perceptual measurement of similar shapes. For example, similar moments do not guarantee similar shapes.

**Edge Histogram Descriptor:** Edge Histogram Descriptor (EHD) [34] captures spatial distribution of edges in an image. This descriptor is like Color Layout Descriptor which does the same by capturing colors. EHD is useful in image matching when the texture of image is not homogenous. It initially divides the input (gray scale) image into some sub-blocks, then for each of them computes local edge histogram. In this regard, edges are categorized into five groups: vertical, horizontal, 45 and 135 diagonals and isotropic (non-directional). Five bins are assigned to these edges. Total bins in sub-images form the actual descriptor [34].

**Histogram of Oriented Gradients**: Histogram of Oriented Gradients (HOG) is a common descriptor, which is used in computer vision and image processing, especially for object detection. Dalal et al. [14] introduced a human detection algorithm using HOG. They designed this descriptor by simulating the procedure of visual information processing in the brain. HOG is robust to local changes of appearance, and position. For computing HOG features, orientation histograms of edge intensity in a local region are considered. This technique counts occurrences of gradient orientations

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

in local segments of an image.

HOG descriptor is similar to some methods such as edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts. But for improving accuracy, HOG is calculated on a dense grid of uniformly spaced cells. Furthermore, it uses overlapping local contrast normalization. The HOG descriptor has some advantages over other description methods. Since the HOG descriptor operates on localized cells, the method is relatively invariant to local geometric and photometric transformations except for object orientation. Such changes would only appear in larger spatial regions. Therefore, the HOG is particularly an appropriate descriptor for human detection in images [14].

- **Texture Descriptors**

Textures are an important part of our life, they can define quality of a particular object or concept, such as the fur of bears, the streakiness of grass and the roundness of pebbles. Technically, images that contain a certain kind of pattern are interpreted as textures. They have often a very uniform (homogeneous) structure, although this is not always the case. Texture is one of the prominent primitives in human vision. Generally, texture features are used to identify or describe contents of images, such as clouds, bricks, hair and so on. These descriptors are designed to describe texture patterns in images. Naturally, they are appropriate for texture images.

**Gabor Filters:** Gabor filters transform [35] is a good multi-resolution approach to represent the texture of an image. It can be computed in an effective way using multiple orientations and scales. Because of its spatial property, which is similar to mammalian perceptual vision, researchers widely use this method in image processing. Since Gabor filters have multiple orientation approach, they are proven to perform better than wavelet transforms and other multi-resolution approaches in representing the texture of an image.

- **Compact Composite Descriptors**

Compact Composite Descriptors (CCD) include a group of low-level features for describing various types of multimedia information. In fact, the CCD approach enables us to reduce the size of image features without affecting the visual content of the image. Two descriptors in CCD family are Color and Edge Directivity Descriptor (CEDD), and Fuzzy Color and Texture Histogram (FCTH). They combine color and texture information in order to describe the visual content of natural color images.

**Color and Edge Directivity Descriptor**: Color and Edge Directivity Descriptor (CEDD) [36] incorporates color and texture information in a histogram. Its length does not exceed 54 bytes per image. This descriptor is suitable for using in large image databases. The low computational need for extraction process is one of the most important benefits of the CEDD, in comparison with most of MPEG-7 descriptors.

For color information, it uses two fuzzy systems to reduce the scale of the colors of the image into 24 colors. Moreover, to extract texture information, CEDD exploits a fuzzy version of five digital filters, which are introduced by the MPEG-7 EHD and form 6 texture areas [36].

**Fuzzy Color and Texture Histogram:** Fuzzy Color and Texture Histogram (FCTH) [37] is a

combination of 3 fuzzy systems. FCTH size is limited to 72 bytes per image. Similar to CEDD, this descriptor is also appropriate for large image databases. This feature could be applied to retrieve images accurately even in distorted situations such as deformations, noise and smoothing.

FCTH and CEDD use the same color information. Also they extract texture information by utilization of the high frequency bands of the Haar wavelet transform in a fuzzy system where this transform forms 8 texture areas [37].

### b. Clustering and key-frame selection

In our method, for key-frame extraction, we index all frames and employ a clustering algorithm, which assigns each frame to one of n clusters. Here n is a fixed number. We choose k-medoid, which is a very common partitioning algorithm, to cluster frames. This technique is similar to well-known k-means clustering. k-medoid has two main advantages which make it appropriate for key-frame selection. First, the cluster centers are always one of the data points, while k-means uses artificial cluster centers. Moreover, clustering is not based on features themselves or the feature space and only depends on the dissimilarity measure that is applied to the image feature vectors.

Finally, the resulting n clusters group frames that are visually similar. This grouping is according to the chosen image representation. The clusters' medoids $M_1, M_2, \dots M_n$ have the minimum distance to all elements of their clusters. Therefore, they are referred as the most descriptive elements of their group. Furthermore, we rank the chosen key-frames relative to their capability to describe the content of the video sequence. In fact, medoid of the largest cluster summarizes the major part of the video. In other words, more of the video's duration is covered by this cluster. Also the medoid of the smallest cluster summarizes a shorter part of the video.

## 3.1.2 Selection of segments of input video

Video over time is a 3D cube of data, in x, y and t dimensions. The t dimension is called the temporal size of a video. Since we employ local spatio-temporal features to recognize human actions, we should preserve temporal aspect of input video. After key-frame extraction, we construct a new input video by using the neighbor frames of each key-frame to form continuous segments and join them together. Unsupervised feature learning algorithms capture information from small regions of video which are called 3D input patches. We realize that for an efficient learning these patches should not have any discontinuity in temporal dimension. Therefore, we have proposed that the number of key-frames' neighbor frames, which makes the length of a continuous segment should be equal to the temporal size of video patches.

The Highlighted video clip $\mathcal{K}$ is defined as follows:

$$\mathcal{K} = \mathcal{A}_{\text{Highlighted video clip}}(V) = E_{i_1} \odot E_{i_2} \odot \dots \odot E_{i_k} \qquad (4)$$

Where $\mathcal{A}_{\text{Highlighted video clip}}$ denotes the procedure of regenerating video clip. $E_i \sqsubset V$ is the ith excerpt to be included in the highlighted video clip, and $\odot$ is the excerpt assembly and integration operation includes cut, fade, dissolve and wipe. In this paper, the cut operator is used to concatenate video segments.

To clarify our proposed method, we present an example: suppose that a video clip has 63 frames. After key-frame extraction process, three frames: 46, 2 and 60 are selected as key-frames. In second step we construct continuous segments. In this example, the length of each segment would be 14. For the first key-frame (42nd frame) by doing left and right padding, an excerpt which includes frames 39 to 52, is obtained. For other two key-frames, we cannot do left and right padding equally. By including neighbors of second key-frame (2nd frame), second excerpt is obtained. This excerpt includes frames 1 to 14. And finally for third key-frame (60th frame), the last excerpt

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

includes frames 50 to 63. The new video clip with 42 frames is constructed by concatenating these three obtained excerpts.

### 3.2 Human Action Recognition Pipeline

Recently, bottom-up approaches have become more popular in the Human Action Recognition, especially in unconstrained settings. Bag- of-Words (BoW) framework is a widely used approach.

In the standard BoW framework, first of all the salient features are discovered at spatio-temporal scales from all training samples. Then, they are combined to form a single action dictionary/codebook. The elements of this dictionary are called primitives, or attributes, or prototypes, or visual words. We will use the term visual word. Conventionally, for the dictionary learning a standard vector quantization (VQ) such as k-means is employed. K-means algorithm groups features with the similar motion and appearance patterns in the same cluster, which is corresponding to a visual word. For encoding a video content, features are assigned to the closest cluster. Then, an action is described by a single representation xS. This representation is the L1-normalized term frequency occurrences of the features in the whole video over the dictionary of visual words. The standard single action representation provides a global representation of the video content. Finally, multi-class SVM is used to classify human actions.

In this paper, we follow the same pipeline as Wang et al. [12]. They have used descriptors such as HOG3D, to extract features from videos on a dense grid in which cube samples overlap 50% in x, y and t dimensions. We use stacked convolutional ISA to learn features. Stacked convolutional ISA is introduced by Le et.al [18] as an unsupervised feature learning method. Also, k-means vector quantization and $\chi^2$-kernel SVM are used in this pipeline.

#### Stacked Convolutional ISA

Independent Subspace Analysis (ISA) can be considered as a generalization of Independent Component Analysis (ICA). Both of these algorithms are well-known in the field of natural image statistics [38, 39]. Experimental studies about these algorithms have shown that they can learn acceptable fields similar to the V1 area of visual cortex when applied to static images and the MT area of visual cortex when applied to sequences of images [38, 40]. A benefit of ISA, in comparison to the more standard ICA algorithm, is that it learns features, which are robust to local translations. A drawback of ISA, as well as ICA, is that its training can be very slow when input dimensions are very high [18].

We can describe an ISA network as a two-layered network. In the first layer, the weights W are learned and in the second layer the weights V are fixed. These two layers called simple units and pooling units, respectively. Given an input pattern $x^t$, the activation function of each second layer unit is

$$p_i(x^t; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} \left(\sum_{j=1}^{n} W_{kj} x_j^t\right)^2} \qquad (5)$$

Also, parameters W are defined by solving following equation. ISA learns these parameters through finding sparse feature representations in the second layer [18].

$$\underset{W}{\text{minimize}} \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V) \qquad (6)$$

$$\text{subject to} \quad WW^T = I$$

Where, $W \in R^{k \times n}$ is the weight of first layer and $V \in R^{m \times k}$ is the weights of second layer ($V$ is typically fixed). Here, $n$ is input dimension, Moreover $k$ and $m$ are number of simple units and pooling unit respectively. Also, in this equation $\{x^t\}_{t=1}^{T}$ are whitened input examples [18].

Le et al. [18] scaled up the original ISA to larger input data such as video sequences. They employ two principal concepts from convolutional neural networks [41]: convolution and stacking. In detail, features are learned with small input patches. Then, the learned features are convolved with a larger region of the input data. The achieved outputs of convolution are inputs to the layer above. By applying convolutional stacking idea, the algorithm can learn a hierarchical representation of the data which is appropriate for recognition task [42]. Finally, features from both layers are combined together and used as local features for classification.

When we have video stream, the inputs to this network are 3D video blocks instead of image patches. More specifically, a sequence of image patches are flattened into a vector. This vector becomes input feature to the ISA network. In this model, input patches to the first layer have 16x16 spatial and 10 temporal sizes. And in the second layer input patches have the spatial size of 20x20 and temporal size of 14. A sample of 3D patch as input to ISA network is illustrated in Figure 4.

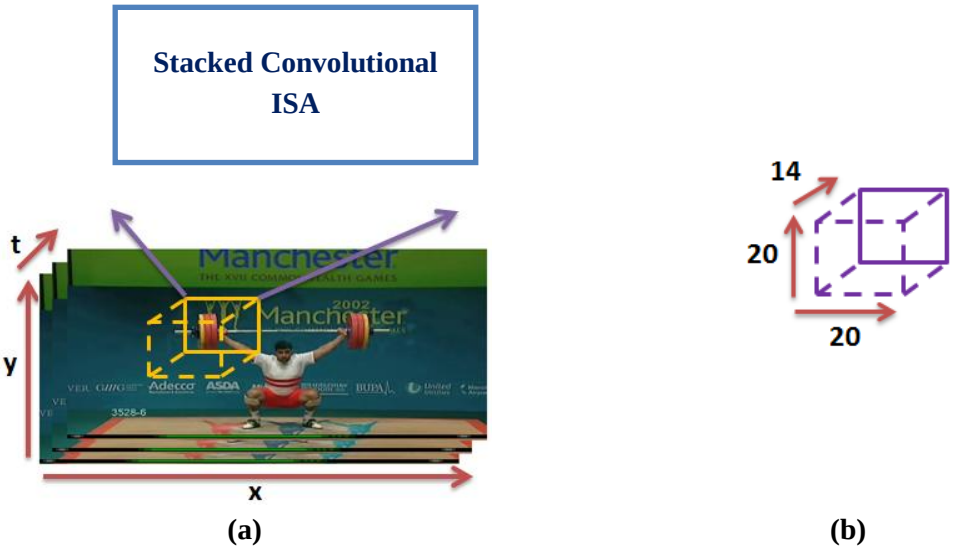


**Figure 4** Stacked Convolutional ISA for video data. (a) The inputs to Stacked Convolutional ISA network are 3D video blocks. (b) An input patch with size 20x20 (spatial) and 14 (temporal).

## 4. Experimental Setup

In this section we will describe the datasets used for our experiments as well as the evaluation procedure used to assess our results. We have evaluated the impact of different low-level features and number of key-frames in key-frame extraction process for the action classification task and employed the evaluation measures proposed by the authors of the datasets.

## 4.1 Datasets

We have performed our experiments on two different action datasets: KTH [43] and UCF sports action [44]. These datasets can be downloaded from original authors' websites.

**KTH action dataset:** In the literature, The KTH dataset [43] is known as a large set with strong intra-subject variations which is commonly used to evaluate action recognition mechanisms and various results have been reported. This dataset includes six human actions: running, boxing, walking, jogging, hand waving, and hand clapping with about 600 choreographed video samples. Each action class is performed several times by Twenty-five subjects. The sequences were recorded under 4 different conditions: outdoors (S1), outdoors with scale change (fast zooming in/out) (S2), outdoors with different clothes (S3), and indoors (S4). The background is homogeneous and static

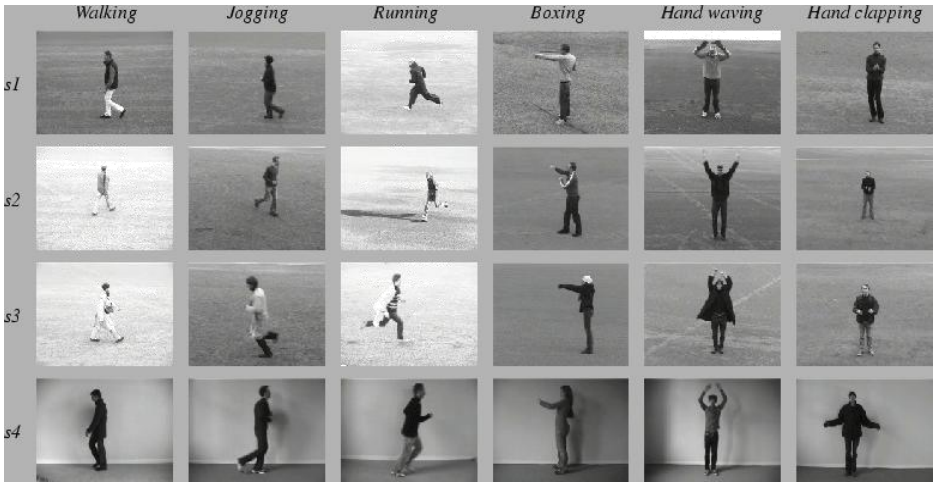


**Figure 5** Sample frames of KTH action dataset.

in most sequences. Each clip lasts between 10 to 15 seconds and is sampled at the rate of 25Hz with an image frame size of 160 by 120 pixels. In our experiments, we have used the original experimental setup of the authors, i.e., the video samples are divided into a test set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and a training set (the remaining 16 subjects). For evaluation, we train a multi-class SVM and report average accuracy over all classes as a performance measure (Figure 5).

**UCF sports dataset:** The UCF Sport actions dataset [50] contains ten different types of human actions such as swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking. The dataset consists of 150 video samples with a large intra-class variability, which are collected from the Youtube website. This dataset is challenging due to diverse ranges of views and scene changes with moving camera, clutter, and partial occlusion. To increase the amount of data samples, A horizontally flipped version of each video is also used during training to increase the data samples [7]. We use a leave-one-out approach in a way that the algorithm is trained on all the sequences together with their flipped versions except for a selected instance, and its flipped version, which would be used in the testing phase (Figure 6).


**Figure 6** Sample frames of UCF-Sport dataset.

### 4.2 Evolution framework

Bag-of-Words (BoW) framework produces a video sequence, which is represented as a bag of local spatio-temporal features. We use the identical pipeline which introduced by Wang et. al [12]. In the first stage of this pipeline, local features are extracted, and then vector quantization by k-means is applied on extracted features. After representing video data, we use a non-linear support vector machine [45] with a $\chi^2$-kernel for the classification task. For multi-class classification, we employ the one-versus-all approach and select the class label with the highest score. In order to increase precision, this pipeline proposes to initialize k-means eight times with different values for centers and reports the one with the lowest error. Findings of Le et al. [18] led us to use higher-level features that are learned by stacked convolutional ISA instead of hand-designed features. In comparison to the basic pipeline, the only difference is that we added a stage for preparing input video in order to highlight its important data and remove redundant or less informative video segments. Then, the prepared video clip is used as input to the pipeline.

## 5. Experimental Results and Discussions

This section presents our experimental results for various low-level features and also different number of key-frames in the key-frame extraction process. In our experiments, the length of input sequence is equal to the number of key-frames times the temporal size of the sequence. ISA extracts features from input patches, which have spatial size of 20x20 and temporal size of 14. For example, 42 frames would be used as input when the number of key-frames is three ($3 \times 14 = 42$). Moreover, we have exploited different kinds of image descriptors, such as color descriptors. Some common color descriptors are color Histogram, Auto Color Correlogram (ACC), Color Layout Descriptor (CLD) and Scalable Color Descriptor (SCD). Also, Gabor filters are used as texture descriptor. Other types of applied descriptors are Compact Composite Descriptors (CCD), which are Color and Edge Directivity Descriptor (CEDD) and Fuzzy Color and Texture Histogram (FCTH). Finally, we have employed shape descriptors, like Edge Histogram Descriptor (EHD) and Histograms of Oriented Gradients (HOG). These descriptors are explained in Section 3.1.1. The results of the

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

different datasets are described in Sections 5.1 and 5.2.

Due to large memory requirements for loading video clips and featuring vector quantization by k-means, we would subsample the original UCF sequences to half spatial resolution in all of our experiments. This can enable us to compare all methods on the same data. Note that due to random initialization of k-means which used for codebook generation, we have a standard deviation of approximately 0.5% in our experiments.

## 5.1 KTH actions dataset

A summary of the KTH dataset's properties, which is obtained from our analysis is demonstrated in Table1.

As it can be seen in Table 1, since KTH dataset is recorded under constrained conditions, it should be considered as a collection of artificial video sequences. Therefore, in this dataset we have more sufficient video samples with numerous frames. Moreover, Because of periodic form of performed actions in the dataset, it contains redundant information, and we might use more information than required when performing the recognition task. Thus, we can correctly recognize actions from very short sequences.

To clarify the problem, we present an example shown in Figure 7: consider a sample of "boxing"

**Table 1** properties of KTH action dataset

| **KTH Properties** |
|---|
| Constrained Video data |
| Rather artificial actions |
| Homogeneous and static background |
| Periodic |
| Grayscale |
| 599 video instances |
| 6 Action Classes |
| No. of samples per Class: 64 train and 36 test |
| Min No. of frames per instance: 204 |
| Max No. of frames per instance: 1492 |
| Mean No. of frames per instance: 483 |

class which has 428 frames. We first extract three key-frames (circulated number in Figure 7). Then, three segments are constructed by adding neighbor frames of those key-frames. According to temporal size of input patches to ISA, the length of each excerpt would be 14. Finally, we concatenate these excerpts and generate a new input video with 42 frames, which is a video clip with significantly shorter duration.

We have employed various types of image descriptors for key-frame extraction and different number of key-frames on KTH action dataset. Results are demonstrated in Table2.

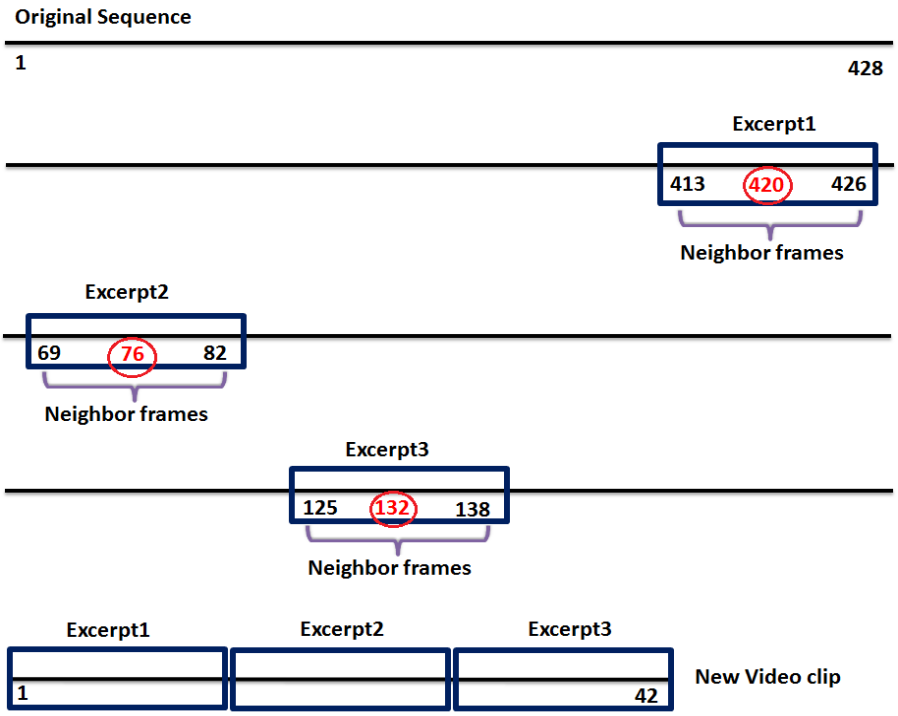


**Figure 7** An example of constructing new input video clip. An instance of "boxing" class from KTH dataset.

**Table 2** Accuracy of different image descriptors (used in key-frame extraction process) with different number of key-frames for KTH action dataset.

| **Image Descriptor (Low-level feature)** | | ***Three key-frames*** | ***Five key-frames*** | ***Seven key-frames*** | ***Nine key-frames*** | ***Means and standard deviations*** |
|---|---|---|---|---|---|---|
| **Color** | **Color Histogram** | 88.6 | 88.9 | 88.5 | 89.7 | 88.9 ±0.54 |
| | **ACC** | 89.9 | 89.9 | 90.4 | 90.0 | 90.1 ±0.24 |
| | **CLD** | 88.7 | 88.5 | 88.9 | 89.5 | 88.9 ±0.43 |
| | **SCD** | 88.1 | 88.1 | 88.7 | 88.8 | 88.4 ±0.38 |
| **Texture** | **Gabor** | 88.7 | 88.8 | 89.3 | 89.9 | 89.2 ±0.55 |
| **Compact Composite (CCD)** | **CEDD** | 88.8 | 88.4 | 88.9 | 89.4 | 88.9 ±0.41 |
| | **FCTH** | 89.0 | 88.5 | 88.7 | 88.7 | 88.7 ±0.21 |
| **Shape** | **EHD** | 88.6 | 88.9 | 88.5 | 89.7 | 88.9 ±0.54 |
| | **HOG** | **92.7** | **93.7** | **94.05** | **94.05** | **93.6 ±0.64** |
| | **Original Samples** | - | - | - | - | 91.4±0.50 |

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

As indicated in Table 2, the accuracy of achieved results is almost the same, when we use original sequences for all of the frames. We have slightly decreased accuracy in all cases, except for the one which uses HOG descriptor. Accuracy of the original video clips equals to 91.4 when we use them as input, and it is obtained with dense sampling. Dense features capture different types of motions.

Generally, the superiority of low-level features differs for different video data. The procedure of video summarization is highly domain dependent and may not perform the same for more general cases. There is a different combination of low-level features that performs the best for the aim of key-frame selection for each dataset [46]. As we can see in Table 1 for KTH, color and texture descriptors and even their combination, and shape descriptors such as EHD, lead to select acceptable key-frames, which are not the best key-frames for this dataset. Consider the fact that the video sequences are grayscale and their background is mostly homogeneous and static. Also note that our purpose of key-frame extraction is to select excerpts of video clip, which are more informative for action recognition task, as well as eliminating redundant or unnecessary information. Therefore, we have exploited HOG descriptor, which is an appropriate descriptor for object recognition and human detection [14, 47]. Using this descriptor in key-frame extraction process, yields better key-frames for KTH action dataset.

It can be seen from Table 2 that increasing the number of key-frames, which creates video clips with longer durations, does not necessarily provide more informative data for recognition task. Moreover, for different number of selected key-frames, some descriptors work better than the others.

## 5.2 UCF sports dataset

Similar to what we have done for KTH, in the first step we analyze different aspects of UCF dataset. A summary of its properties is provided in Table 3 and Table 4.

**Table 3** Properties of UCF sports dataset.

| **UCF Sports Properties** |
|---|
| Uncontrolled Video data |
| Realistic actions |
| Non-homogeneous background |
| Non-periodic |
| Colorful |
| 150 instances +150 Horizontally Flipped Version |
| 10 Action Classes |
| No. of instances per Class: 6 to 22 |
| Min No. of frames per instance: 20 |
| Max No. of frames per instance: 145 |
| Mean No. of frames per instance: 62 |

**Table 4** Properties of every action class in UCF sports dataset.

| Class number | No. of Samples per Class | Mean No. of Frames per Sample |
|---|---|---|
| C1 "Golf-swing" | 18 | 60 |
| C2 "Kicking" | 20 | 23 |
| C3 "Lifting" | 6 | 107 |
| C4 "Horse- Riding" | 12 | 56 |
| C5 "Running" | 13 | 65 |
| C6 "Skateboarding" | 12 | 70 |
| C7 "Swing-bench" | 20 | 50 |
| C8 "Swing-highbar" | 13 | 72 |
| C9 "Walking" | 22 | 95 |
| C10 "Diving" | 14 | 55 |

As stated in Table 3 and Table 4, UCF-Sport is an uncontrolled video dataset that is recorded in real-world conditions. Collecting samples under these conditions is a rather difficult task; because there are a small number of samples in each class and they do not have many frames. For example, there are just about 460 frames for "Kicking" class (20 samples × 23 frames per sample on average). Such small sample size forces us to use available information optimally.

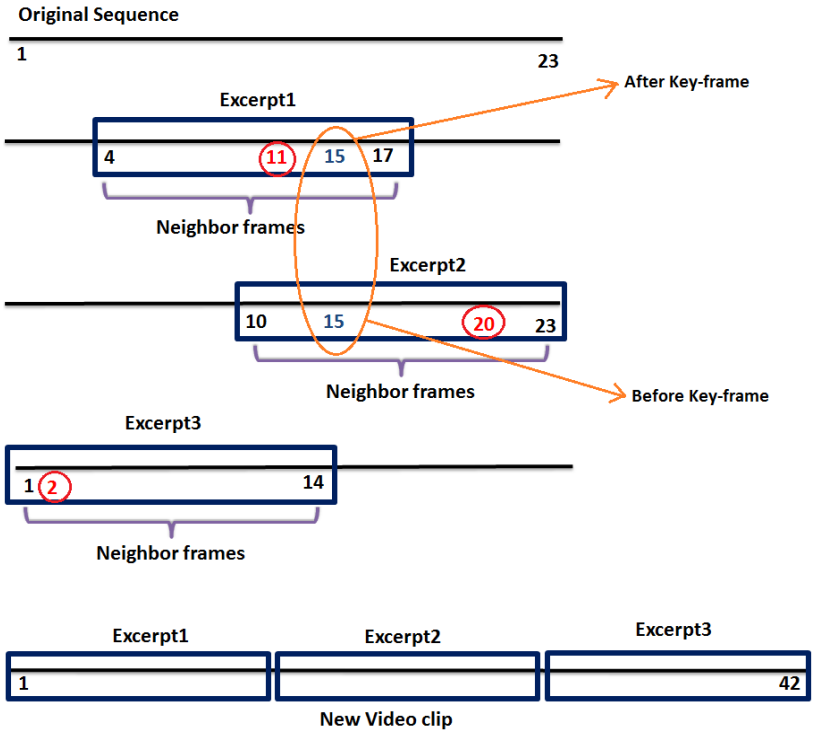


**Figure 8** An example of constructing new input video clip. An instance of "kicking" class from UCF sports dataset.

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

The example shown in figure 8, helps us to clarify this issue. We pick a sample of "Kicking" class with 23 frames. At first, we select three key-frames via mentioned key-frame extraction algorithm (illustrated with circulated numbers in Figure 8). Then we construct three excerpts, where according to temporal size of input patches to ISA, each one has 14 frames and finally by concatenating these three excerpts, we generate a new sample of 42 frames video segment. Note that as occurred in this example, it is possible that the length of the new video clip be longer than the original sequence. It happens when some excerpts have frames in common with others. In fact, there could be multiple copies of a frame in our new video clip. For instance, 15th frame exists in two excerpts: in the first excerpt it appears after the key-frame and in the second one it appears before the corresponding key-frame. Although 15th frame is spatially identical in two segments, it appears in different temporal position. In an attempt to solve the problem of little data, we have used more informative frames optimally. Since ISA extracts spatio-temporal features from input patches, any discontinuity or duplication in each input patch will defect the temporal aspect of patches. Therefore, the extracted features will not be efficient and the performance of the method will be decreased.

We have applied various low-level features to key-frame extraction with different input lengths, which are dependent to the number of key-frames. Results are represented in Table 5.

**Table 5** Accuracy of different image descriptors (used in key-frame extraction process) with different number of key-frames for UCF sports dataset.

| **Image Descriptor (Low-level feature)** | | ***Three key-frames*** | ***Five key-frames*** | ***Seven key-frames*** | ***Means and standard deviations*** |
|---|---|---|---|---|---|
| **Color** | **Color Histogram** | 83.1 | **87.2** | **87.2** | 85.8 ±2.37 |
| | **ACC** | **87.5** | **88.0** | **88.5** | **88.0 ±0.50** |
| | **CLD** | 85.9 | 86.2 | 85.9 | 86.0 ±0.20 |
| | **SCD** | 85.9 | 86.2 | 85.9 | 86.0 ±0.20 |
| **Texture** | **Gabor** | 83.0 | 85.6 | **86.7** | 85.1 ±1.90 |
| **Compact Composite (CCD)** | **CEDD** | 84.0 | **86.6** | **88.0** | 86.2 ±2.03 |
| | **FCTH** | 84.8 | 85.7 | **87.5** | 86.0 ±1.37 |
| **Shape** | **EHD** | 83.1 | **86.7** | **88.2** | 86.0 ±2.62 |
| | **HOG** | 85.7 | **87.2** | **87.3** | **86.7 ±0.90** |
| | **Original Samples** | - | - | - | 86.5±0.50 |

As represented in Table 5, since the samples are colorful and have non-homogeneous background, applying either of color, texture or shape descriptors, or combination of them results in better performance on KTH dataset. Compared with the original sequence, in most cases we have achieved higher accuracy. For this dataset, ACC performs better than other descriptors.

For each data set there is a descriptor, which has optimal performance, and it is expected to perform the same on similar datasets. Dense features capture background which may provide useful context information. Scene context indeed may be helpful for sport actions which often involve specific equipment and scene type, as illustrated in Figure 6.

### 5.3 Computational complexity

By highlighting input video clips, overall input data for standard action recognition pipeline, is reduced. According to standard action recognition pipeline, some steps such as learning dictionary by vector quantization (VQ) are highly time-consuming. While there is less input data, these steps perform faster. But we should not forget the overhead of computing global features for selecting key-frames in our proposed stage. As a result by adding video highlighting stage, run-time complexity of our method and the original framework are approximately the same. However our proposed method complexity depends on the number of key-frames and selected low-level feature for key-frame extraction process.

In standard pipeline, feature vector quantization by k-means has large memory requirements. Since our proposed method, reduces input data, memory requirement in this step is decreased.

## 6. Conclusion and Future Work

We have explored the effect of highlighting parts of input video data for standard BoW action recognition pipeline. In this paper, we proposed a novel technique in order to achieve this goal. First, key-frames are automatically extracted and then by including neighbor frames, continuous segments are constructed. Finally, by concatenating these segments we generate a new video clip, which is used as input for action recognition framework. In addition, our method is also generalized to be able to perform on real-world settings, where there is small sample size video data such as sport broadcasts. We have examined our method on two benchmark datasets: KTH and UCF-sport. From these two various datasets, we found that analyzing different aspects of a dataset would help to choose appropriate low-level features according to its attributes. Furthermore, in general raw data does not necessarily result in more precision. Selecting more informative parts of data and eliminating unimportant or redundant segments can be useful for recognition task. In addition, this approach can be effective for datasets with either information redundancy or small number of samples.

The advantage of our method is its simplicity and also its capability to work on different datasets. In future, we intend to extend our proposed method to other domains, which suffer from high dimensions of video input as well as (Real-time) online applications. In addition, as a future research we intend to find appropriate key-frame extraction method for each class of action. Our current approach is to provide a new intelligently composed version of a video and treat different classes of actions similarly. This work can increase performance; however it needs extra effort and can be a complex process.

**ISSN 2292-1028(Print) 2292-1036(Online)**
**DOIC: 2292-1036-2014-03-001-32**

## References

[1]. Poppe, R. (2010),"A survey on vision-based human action recognition",*Image and Vision Computing*, (6):976–990.

[2]. Hu, W., Tan, T., Wang, L., Maybank, S. (2004),"A survey on visual surveillance of object motion and behaviors", *IEEE Transactions on Systems, Man and Cybernetics*, 34: 334-352.

[3]. Hu, W., Xie, N., Li, L., Zeng, X., Maybank, S. (2011),"A survey on visual content-based video indexing and retrieval", *IEEE Transactions on Systems, Man, and Cybernetics*, 41 (6):797–819.

[4]. Rodriguez, M. (2010),"CRAM: Compact representation of actions in movies", *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 2010, 3328–3335.

[5]. Aggarwal, J., Ryoo, M. (2011),"Human activity analysis: A review", *ACM Computing Surveys*. 1–47.

[6]. Truong, B.T., Venkatesh, S. (2007),"Video abstraction: A systematic review and classification", *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1).

[7]. Weinland, D., Ronfard, R., Boyer, E. (2011),"A survey of vision-based methods for action representation, segmentation and recognition", *Computer Vision and Image Understanding*, (2): 224–241.

[8]. Zouba, N., Boulay, B., Bremond, F., Thonnat, M. (2008), "Monitoring activities of daily living of elderly based on 3d key human postures", *International Cognitive Vision Workshop*, 37–50.

[9]. Laptev, I., Caputo, B., Schuldt, C., Lindeberg, T. (2007),"Local velocity-adapted motion events for spatio-temporal recognition", *Computer Vision and Image Understandin*g,207–229.

[10]. Fergus, R., Perona, P., Zisserman, A. (2003),"Object class recognition by unsupervised scale-invariant learning", *Computer Vision and Pattern Recognition, 2003*, 2: 264.

[11]. Lazebnik, S., Schmid, C., Ponce, J. (2006),"Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", *Computer Vision and Pattern Recognition, 2006*, 2: 2169-2178.

[12]. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C. (2009),"Evaluation of local spatio-temporal features for action recognition",*British Machine Vision Conference, 2009*.

[13]. Lowe, D. (2004),"Distinctive image features from scale-invariant keypoints", *International Journal on Computer Vision, 2004*, 60(2), 91-110.

[14]. Dalal, N., Triggs, B. (2005),"Histograms of oriented gradients for human detection", *Computer Vision and Pattern Recognition, 2005*, 1: 886-893.

[15]. Lee, H., Battle, A., Raina, R., Ng, A. Y. (2007),"Efficient sparse coding algorithms". *Conference on Neural Information Processing Systems, 2007*, 19: 801.

[16]. Hinton ,G., Osindero, S., The, Y. (2006),"A fast learning algorithm for deep belief nets", *Neural Computing*,18(7), 1527-1554.

[17]. Berkes, P., Wiskott, L. (2005). "Slow feature analysis yields a rich repertoire of complex cell properties", *Journal of Vision*, 5(6), 9.

[18]. Le, Q., Zou, W., Yeung, S., Ng, A. (2011), "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis*". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011*, 3361–3368.

[19]. Jhuang, H., Serre, T., Wolf, L., Poggio, T. (2007),"A biologically inspired system for action recognition*", In Proceeding of IEEE Interaction Conference on Computer Vision, 2007*, 1-8.

[20]. Schindler, K., Gool, L.V. (2008), "Action Snippets: How many frames does human action recognition require?",*In IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, 2008*, 1–8.

[21]. Divakaran, A., Peker, K.A., Radhakrishnan, R., Xiong, Z., Cabasson, R. (2003),"Video summarization using Mpeg-7 motion activity and audio descriptors", *Video Mining. US, 2003*, 91-121.

[22]. Bennett, E.P., McMillan, L. (2007),"Computational time-lapse video", *In ACM Transactions on Graphics, 2007*, 26(3): 102.

[23]. Carlsson, S., Sullivan, J. (2001). "Action recognition by shape matching to key frames", *Workshop on Models versus Exemplars in Computer Vision, 2001*, 1:18.

[24]. Zhu, X., Wu, X., Fan, J.,Elmagarmid, A. K., Aref, W. G.(2004),"Exploring video content structure for hierarchical summarization", *Multimedia Systems*, 10(3): 98–115.

[25]. Kasamwattanarote, S., Cooharojananone, N., Satoh, S., Lipikorn, R. (2010),"Real time tunnel based video summarization using direct shift collision detection", *In Advances in Multimedia Information Processing*, 6297: 136–147.

[26]. Kim. C., Hwang. J. (2000),"An integrated scheme for object-based video abstraction", *In ACM Multimedia*, 303–311.

[27]. Farin, D., Effelsberg, W., deWith, P. H. N. (2002),"Robust clustering-based video summarization with integration of domain-knowledge", In IEEE International Conference on Multimedia and Expo, 2002, 89–92.

[28]. Hu, Y., Zheng, W. (2011),"Human Action Recognition Based on Key Frames", *Advances in Computer Science and Education Applications Communications in Computer and Information Science*, (202): 535-542.

[29]. Chang, H.C., Yang, C.K. (2012),"Fast content-aware video length reduction", *Signal, Image and Video Processing*, 1-15.

[30]. Avidan, S., Shamir, A. (2007),"Seam carving for content-aware image resizing",*In ACM Transactions on graphics*, 26(3):10.

[31]. Huang, J., Kumar, S. R., Mitra, M., Zhu, W., Zabih, R. (1997),"Image indexing using color correlogram", *In Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition, 1997*, 762-768.

[32]. Kasutani, E., Yamada, A. (2001),"The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval", *International Conference in Image Processing, Proceedings*, 1:674-677.

[33]. Ohm, J.R., Cieplinski, L., Kim, H. J., Krishnamachari, S., Manjunath, B. S., Messing, D. S., Yamada, A. (2003). "The MPEG-7 Color Descriptors".

[34]. Park, D. K., Jeon, Y. S., Woon, C. S. (2000),"Efficient use of local edge histogram descriptor", *In Proceedings of the 2000 ACM workshops on Multimedia, 2000*, 51-54.

[35]. Manjunath, B., Ma, W. (1996), "Texture features for Browsing and retrieval of image data",

*IEEE transactions on pattern analysis and machine intelligence*, 18(8):837-842.

[36]. Chatzichristofis, S. A., Boutalis, Y. S. (2008),"CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval", *In Computer Vision Systems* .312-322.

[37]. Chatzichristofis, S. A., Boutalis, Y. S. (2008), "Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval", *Ninth International Workshop on Image Analysis for Multimedia Interactive Services, 2008*, 191-196.

[38]. Hyvarinen, A., Hurri, J., Hoyer, P. (2009). "Natural Image Statistics".

[39]. Van Hateren, J., Ruderman, D. (1998), "Independent component filters of natural images compared with simple cells in primary visual cortex", *Proceeding of Royal Society*, 265(1394), 359-366.

[40]. Van Hateren, J., Ruderman, D. (1998),"Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex", *Proceedings of the Royal Society: Biological Sciences*, 265(1412), 2315-2320.

[41]. LeCun, Y., Bengio, Y. (1995),"Convolutional networks for images, speech, and time- series", *The Handbook of Brain Theory and Neural Networks*, 3361.

[42]. Lee, H., Grosse, R., Ranganath, R., Ng, A. (2009),"Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", I*n Proceedings of the 26th Annual International Conference on Machine Learning, 2009*, 609-616.

[43]. Schuldt, C., Laptev, I., Caputo, B. (2004). "Recognizing human actions: A local SVM approach", *In Proceedings of the 17th International Conference on International Conference on Pattern Recognition, Cambridge, UK, 2004*, 3: 32-36.

[44]. Rodriguez, M., Ahmed, J., Shah, M. (2008),"Action match: A spatio-temporal maximum average correlation height filter for action recognition", *In IEEE Conference on Computer Vision and Pattern Recognition, Alaska, 2008*, 1-8.

[45]. Chang, C. C., Lin, C. J. (2001). "LIBSVM: a library for support vector machines".

[46]. Lux, M., Schoffmann, K., Marques, O., Boszormenyi, L. (2009),"A novel tool for quick video summarization using key-frame extraction techniques". *In Proceedings of the 9th Workshop on Multimedia Metadata CEUR Workshop Proceedings, 2009*, 441:19-20.

[47]. Kobayashi, T., Hidaka, A., Kurita, T. (2008),"Selection of histograms of oriented gradients features for pedestrian detection", *In Neural Information Processing*, 598-607.