Rahman *et al. EURASIP Journal on Image and Video Processing* (2017) 2017:74
DOI 10.1186/s13640-017-0221-2

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Exploiting textures for better action recognition in low-quality videos

Saimunur Rahman[†], John See[*†] and Chiung Ching Ho

Abstract

Human action recognition is an increasingly matured field of study in the recent years, owing to its widespread use in various applications. A number of related research problems, such as feature representations, human pose and body parts detection, and scene/object context, are being actively studied. However, the general problem of video quality—a realistic issue in the face of low-cost surveillance infrastructure and mobile devices, has not been systematically investigated from various aspects. In this paper, we address the problem of action recognition in low-quality videos from a myriad of perspectives: spatial and temporal downsampling, video compression, and the presence of motion blurring and compression artifacts. To increase the resilience of feature representation in these type of videos, we propose to use textural features to complement classical shape and motion features. Extensive results were carried out on low-quality versions of three publicly available datasets: KTH, UCF-YouTube, HMDB. Experimental results and analysis suggest that leveraging textural features can significantly improve action recognition performance under low video quality conditions.

Keywords: Action recognition, Spatio-temporal features, Shape, Motion, Textures, Low-quality video

1 Introduction

Human action recognition is one of the most important research areas in computer vision due to its usefulness in real-world applications such as video surveillance, human computer interaction, and video archival systems. However, action recognition still remains a difficult problem when dealing with unconstrained videos such as web videos, movie and TV shows, and surveillance videos. There are a wide range of issues, ranging from object-based variations, such as appearance, view pose and occlusions, to more complicated scene-related variations such as illumination changes, shadows, and camera motions [1].

While there is significant amount of progress in solving these problems, the issue of video quality [1, 2] has received much less research attention. The recognition of human actions from low-quality video is highly challenging as valuable visual information is compromised by various internal and external factors such as low resolution, sampling rate, compression artifacts and motion blur, camera jitter, and shake. Figure 1 shows a few sample frames from videos that have been severely compromised in the aspect of quality. Many surveillance systems require further video analysis to be performed on compactly stored video data [3] while mobile devices strive to incorporate high-level semantics into real-time streaming [4]. Therefore, for reasons such as these, action recognition in low-quality videos should be further investigated as it offers new insights and challenges to the research community.

Shape and motion features have recently become popular for their great success in action recognition [5–8]. Existing methods that utilize these features mainly consists of two main steps: *feature detection* and *feature description*. In *feature detection*, important salient points are detected from a video and then a visual pattern surrounding the detected point (often called a "patch") is then described in the *feature description* phase. The quality of detected interest points is highly dependent on the quality of the video as important points may be missed in cases where video quality is poor. Also, shape and motion descriptors such as HOG [9], HOF [5, 6], and MBH [7, 8] becomes less discriminative when the quality of video deteriorates; noisy image pixels can cause gradient and

*Correspondence: johnsee@mmu.edu.my
[†]Equal contributors
Centre of Visual Computing, Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia

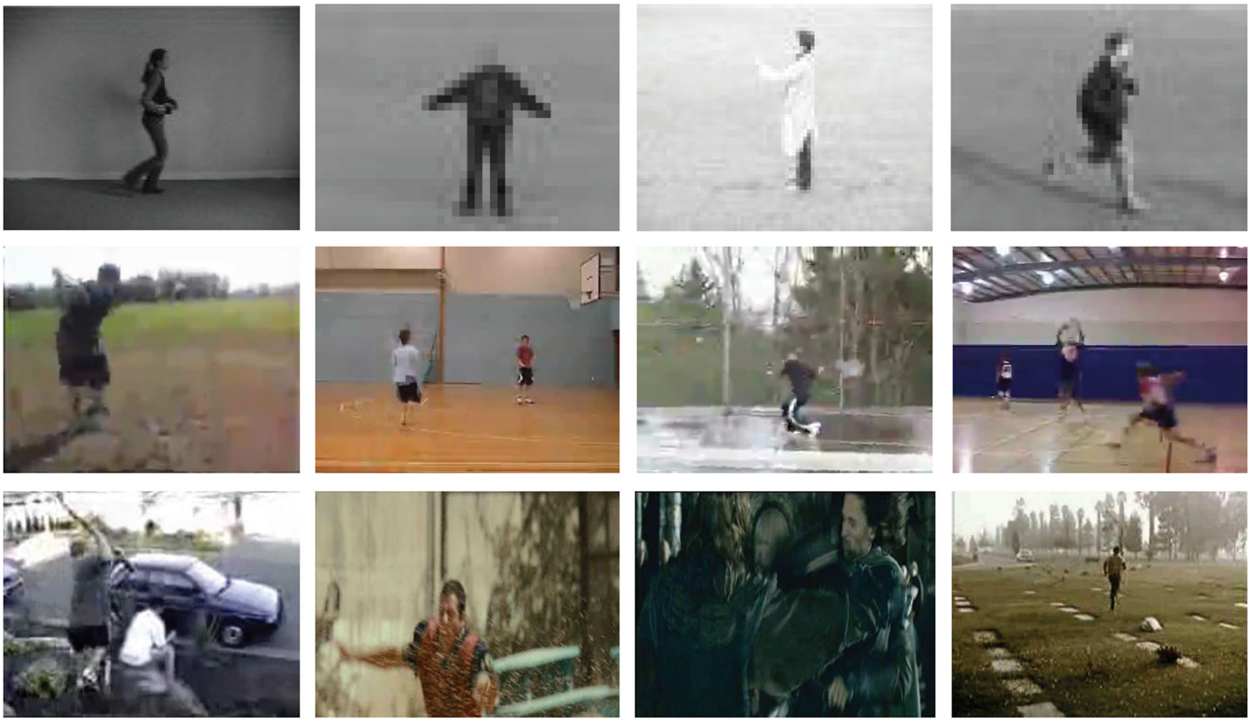

Fig. 1 Sample videos from low-quality subsets of the KTH, UCF-YouTube, and HMDB datasets. The top row are samples from KTH, the middle row are samples from UCF-YouTube, and the bottom row are samples from HMDB. We can observe that the videos are severely compromised by many quality-related factors such as low resolution, lossy compression, and camera motion blurring. As such, the characterization of actions from these videos would be much more challenging

orientation information to be less consistent across action samples of the similar class. Spatio-temporal dynamic textures particularly local binary patterns extended to three orthogonal planes (LBP-TOP) [10] have also been proposed for action recognition [11–13] but they are not as popular or as widely used as shape and motion features. These methods find statistical regularities to describe visual patterns that lie within video frames. Some evaluations [14] observed that textural features, on its own, do not perform consistent enough in videos with complex scenes. This is an expected outcome since the statistical aggregation of patterns is devoid of any spatial or temporal localization. Very few works in literature have particularly address the issue of recognizing actions in low-quality videos; some in the form experimental extensions [15], or to tackle a specific problem such as frame rate reduction [16]. Nevertheless, this is a testament of a potential interest in this issue, but there is presently no systematic investigation into various video-quality-related issues.

In this paper, we attempt to investigate the problem of recognizing human actions from low-quality videos and to uncover how textural features can be used alongside classical shape and motion features to improve the recognition performance under these circumstances. We propose a joint feature utilization framework where *local* shape-motion descriptors obtained from contemporary feature detectors were supplemented by spatio-temporal extensions of *global* texture descriptors. To facilitate the nature of our work, we perform an extensive evaluation on various low-quality versions or quality-oriented subsets of benchmark action recognition datasets: KTH, UCF-YouTube, and HMDB.

The rest of the paper is organized as follows. In Section 2, we delve into some related works in literature while Section 3 introduces our proposed framework and provides a description of the methods employed in our work. We then report and analyze the experimental results in Section 4. Finally, Section 5 concludes the paper and provides future directions.

2 Related works

Vision-based action recognition is a well-studied problem, and many methods [1, 5–8, 17–19] have been proposed in recent years. There are a number of recent survey papers that offer a good overview of related works from the broad, generic scope [20–22] and selected perspectives [23, 24]. Here, we concisely describe related methods from the aspect of their feature selection and representation methods.

2.1 Shape and motion features

Local shape and motion are the most popular features [2, 5–8] for action recognition. A variety of methods that extract shape and motion features have been proposed in literature [5–8, 17, 18]. Inspired by the capability of space-time interest points (STIPs) [25] at capturing local variations, Schüldt et al. [26] use them to extract spatio-temporal local features from action videos. They calculate a multi-scale derivative at center of every local interest point to encode motion information. Their method demonstrated that local motion features are comparatively better than global features. Laptev et al. [27] further improved it by introducing two essential types of local features: shape, in the form of histogram of oriented gradients (HOG), and motion, in the form of histogram of optical flow (HOF). Klaser et al. [28] extended the HOG shape descriptor to three dimensions, i.e., histogram of 3D gradient orientations (HOG3D) [28], which quantizes 3D gradient orientations on regular polyhedrons. However, detection of interest points in [25] solely depends on the spatial quality and temporal fidelity of the video, thus this may be affected if the video quality deteriorates. Also, these mere spatio-temporal extensions do not consider temporal relationship between interest points of subsequent frames. Dollár et al. [17] directly considered the temporal domain in the selection process by proposing a Cuboid detector which selects features surrounding spatio-temporal interest points detected by temporal Gabor filters. The detected cuboids in the video volume are then described by a Cuboid descriptor. Willems et al. [18] use spatio-temporal Hessian Matrix to detect interest points and the extended SURF (ESURF) descriptor for describing features around the detected points. Generally, these methods appear to work well with videos captured in a relatively controlled environment [6] such as KTH [26] action videos.

For videos captured in a relatively complex environment such as the HMDB [29] dataset, interest point based methods such as STIP [25] may sometimes fail to detect important points due to motion clutter from background scenes. To overcome this problem, Wang et al. [6] proposed dense sampling of spatio-temporal video blocks at regular scales and positions and represent them using popular features such as HOG, HOF, and HOG3D. However, the dense sampling strategy is computationally expensive and has a large memory footprint. The authors later proposed Dense Trajectories [7] where densely sampled points are tracked based on the optical flow field. However, feature tracking from dense optical flow fields inadvertently includes camera motion, which may yield less discriminative feature sets. The authors improved their trajectories by performing irrelevant background motion removal using warped flow [8]. Features are then constructed from the trajectories using HOG, HOF, and a robust descriptor called motion boundary histogram (MBH).

2.2 Textural features

The use of textural features is less common in action recognition; a number of notable works are worth mentioning [11–13, 19, 30] but their reported performances were. Kellokumpu et al. [11] first proposed the use of local spatio-temporal texture features for action recognition. They use local binary pattern hon three orthogonal planes (LBP-TOP) [10] to represent an action video in the form of dynamic textures. Their proposed method is capable of capturing the statistical distribution of local neighborhood variations but the holistic nature of extracting these features mean that they may easily be affected by unnecessary background variations and occlusions. Mattivi and Shao [12] proposed to use part-based representations such as interest points to overcome background- and occlusion-related problems. They employed Dollár's feature detector [17] to extract cuboids from video and subsequently, each cuboid is described by extended LBP-TOP (an extension of LBP-TOP to nine slices, three for each plane) descriptor. They also demonstrated that using LBP on gradient images can obtain better performance than using LBP on raw image values, but at the expense of more computations.

Besides directly applying LBP on image frames, there were alternative strategies in literature that used it to extract textures from other forms of images. Kellokumpu et al. [19] used local binary patterns (LBP) to describe motion history and motion energy images which encodes shape and motion information respectively. Ahsan et al. [13] use LBP features to describe mixed block-based directional MHI (DMHI) templates [31].

LBP-based methods are sensitive to noise and illumination changes and they also lack explicit motion encoding. Addressing these issues, Yeffet and Wolf proposed local trinary patterns (LTP) [32] which combined local binary patterns with appearance and adaptability invariance of patch similarity matching approaches. Their method encodes local motion information by taking into account the self-similarity in three neighborhood circles at a particular spatial position. The LTP produces a notably large feature vector, which depends on the number of grid blocks and time slices chosen for a video. More recently, Kataoka et al. [14] investigated thoroughly into the performance of various features of different types (motion, texture, etc.) including texture-based features such as LBP and LTP, on a dense trajectory framework. In their evaluation, they observed that textural features alone do not perform as well as shape or motion features. However, it remains inconclusive as to whether they are more useful under low-quality conditions.

In another new work, Baumann et al. [30] proposed motion binary pattern (MBP) which encodes motion by calculating pixel variations in three consecutive image frames. In order to capture slow and fast motion, they use different time step sizes. Inspired by volume local binary pattern (VLBP) [10] and optical flow, their method produces a very lengthy feature vector and relies heavily on several free parameters that are crucial to its success. So far, they have tested only on the KTH, Weizmann, and IXMAS datasets, which are all relatively smaller than the contemporary datasets used today.

2.3 Action recognition in low-quality videos

There are only a few works in literature that specifically address the impact of low-video quality on action recognition performance; all are limited to only certain factors influencing video quality or done rather ad hocly [15, 16, 33–35].

Chen and Aggarwal [33] proposed to use supervised PCA projected time series of histogram of oriented gradients (HOG) and histogram of oriented optical flow (HOOF) descriptor, to encode pose and motion information of action videos from a far distance. Their work focused on recognizing human actions from a far field of view where the size of humans is typically not more than 40 pixels. A trained classifier is applied to localize action in each image frame, and the features are then computed from these localized coordinates. However, their work only considers the problem of spatial resolution, but did not address other related issues such as camera motion and video compression. The authors also experimented with downsampled frames (with persons as small as 15-pixel tall) and found that the performance deteriorates greatly.

In another work, Reddy et al. [15] conducted a few sensitivity tests involving varying frame rate, resolution (scale), and translation, to test their effect on action recognition performance. Due to the tedious nature of such experimentation, the authors only conducted these tests on the small UT-Tower dataset [33] and not other larger databases. Moreover, the test cases to study scale changes were designed in a rather ad hoc manner. More recently, Harjanto et al. [16] investigated the effects of different frame rates with four popular action recognition methods. A key frame selector was used to select important frames in video. Their evaluation suggests that by selecting a significant amount of important frames, it is still possible to obtain a decent level of recognition accuracy. However, the proposed key-frame selection strategy is solely based on interest points and may not work well if video spatial resolution becomes poor. On the other hand, the work by Ahad et al. [36] focuses solely on the problem of low resolution in activity recognition, but not on low frame rate.

Aside from feature design, a few works [1, 2, 35] focus primarily on the formation of feasible frameworks. They leverage on existing features while crafting the recognition pipeline in an effective way that enhances its performance under low-quality conditions. Our preliminary efforts [1, 2] at exploring the problem of recognizing actions in low-quality video resulted in the establishment of a spatial and temporal downsampling protocol, which provides a systematic procedure for investigating the robustness of methods against decreasing resolution and frame rate. Inspired by the recent breakthrough in deep learning, a recent work [35] incorporated frame-level object features from an ImageNet-trained deep convolutional neural network (CNN) as part of the recognition pipeline, achieving promising results.

Another work by Gao et al. [34] followed the low-resolution downsampling protocol mooted in [2] but using Dempster-Shafer's theory (DS theory) to model activities. Using the KTH and Weizmann datasets, they showed that using DS theory to combine estimated basic belief assignments at each frame can help achieve better performance than popular encoding techniques such as bag-of-visual-words (BoVW) and key-pose modeling. However, they stop short at examining the impact of decreasing video frame rate; their scheme may generalize poorly in such cases since it models the consecutive changes in video.

3 Methods

In this section, at first, we introduce our proposed action framework. Then, we discuss the different components involved such as feature detectors and descriptors.

3.1 Proposed action recognition framework

We propose an action recognition framework based on the shape, motion, and texture features, as illustrated in Fig. 2. The main idea revolves around the utilization of textural information with conventional shape and motion features to improve the recognition of human actions in low-quality videos. In this framework, every input video goes through two distinct extraction steps. In the first step, space-time shape/motion features (derived from interest points or dense trajectories) are extracted by their respective descriptors (i.e., HOG, HOF, MBH). The shape and motion features are then encoded by bag-of-visual-words (BOVW) method [37] to obtain the *local features*. In the second extraction step, spatio-temporal textural features (based on BSIF, LBP, LPQ) are obtained by means of the three orthogonal planes (TOP) extension, forming the *global features*. Finally, both feature vectors are concatenated and a support vector machine (SVM) classifier is used for classification.

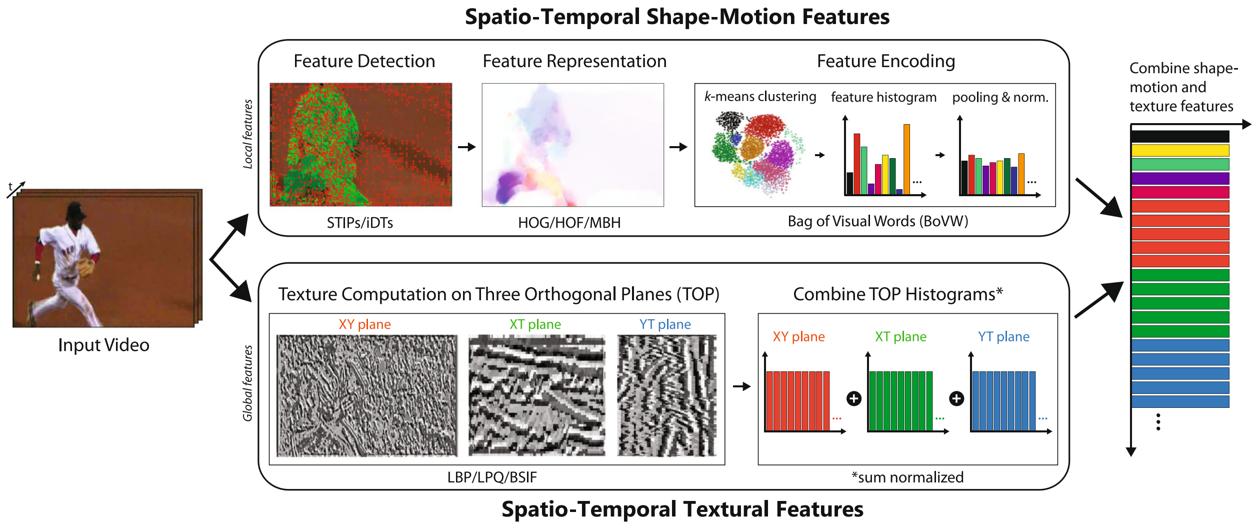

Fig. 2 Proposed joint feature utilization framework for human action recognition

3.2 Shape and motion features

Shape and motion are an expressive abstraction of visual patterns, in space and time respectively. They are critical cues for action recognition, as they are sufficiently invariant to represent commonalities of different instances of a particular action type, while preserving sufficient details in order to differentiate them from different types. To provide a comprehensive coverage of state-of-the-art feature detectors, we employ two different methods that have been widely used in literature: *space-time interest point* (STIP) [5] and *space-time trajectories*, in the form of improved dense trajectories (iDT) [8]. For description, we used the HOG and HOF descriptors in concert [6] for the STIP, and the motion boundary histogram (MBH) for the iDT. These are the most effective descriptors for each detector, as reported in their original works. A brief description of these detectors and descriptors is given as follows:

Space-time interest point: Given an action video, local space-time interest points (STIP) are detected around the location of large variations of image values, which corresponds to motions. Interest points are detected using the Harris3D detector proposed in [5], which is an extension of the popular Harris detector used in image domain [38]. It can detect a decent amount of corner points in space-time domain and is perhaps one of most widely used feature detector for action recognition.

To characterize the shape and motion information accumulated in space-time neighborhoods of the detected STIPs, we applied Histogram of Gradient (HOG) and Histogram of Optical Flow (HOF) feature descriptors as proposed in [26]. The combination of HOG/HOF descriptors produces descriptors of size $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma, \Delta_t(\tau) = 8\tau$ (σ and τ are the spatial and temporal scales). Each volume is subdivided into $n_x \times n_y \times n_t$ grid of cells; for each cell, 4-bin histograms of gradient orientations (HOG) and 5-bin histograms of optical flow (HOF) are computed. We use the original implementation from [5] and standard parameter settings from [6], i.e., $k = 0.0005$, $\sigma^2 = \{4, 8, 16, 32, 64, 128\}$, $\tau^2 = \{2, 4\}$, $\{n_x, n_y\} = 3$ and $n_t = 2$.

Space-time trajectories: Motion information of a video is captured in a dense manner by sampling interest points at an uniform interval and tracking them over a fixed number of frames. To detect space-time trajectories, we used improved dense trajectories (iDT) [8], an extension of the original dense trajectories [7]. A set of points are densely sampled on a grid on eight different spatial scales with a step size of 5 pixels. Points from homogeneous areas are removed by thresholding small eigenvalues of their respective auto-correlation matrices. Tracking of these sampled points are then performed by applying median filtering to the dense optical flow field computed from Färneback's algorithm [39]. Also, static trajectories with lack of motion and trajectories with large displacements due to incorrect optical flow estimation are removed.

In contrast to dense trajectories, iDT is capable of boosting recognition performance by considering camera motions in action videos. It characterizes background motions between two consecutive frames by a homography matrix, which can be calculated by finding similarities between two consecutive frames using SURF [40] and optical flow-based feature matching. After finding feature similarities, RANSAC [41] algorithm is applied to calculate the homography matrix. Based on that, camera motion is removed from video frames before recomputing optical flow. This method known as *warped flow*, results in better descriptor formation with motion estimation that is free from camera motions.

For computational tractability, we use iDT on a single scale in our experiments. We observed that tracking points on multiple spatial scales is computationally very expensive. So, we only track points at the original spatial scale and extract features around its trajectories. Despite that, using a single scale still offers a decent recognition rate (reportedly 2–3% less than multi-scale in [42]). In brief, given an action video V, we obtain N number of trajectories:

$$\mathcal{T}(V) = \{T_1, T_2, T_3, \ldots, T_N\} \tag{1}$$

and, T_n is the n-th trajectory at original spatial scale,

$$T_n = \left\{ \left(x_1^n, y_1^n, t_1^n\right), \left(x_2^n, y_2^n, t_2^n\right), \left(x_3^n, y_3^n, t_3^n\right), \ldots, \left(x_L^n, y_L^n, t_L^n\right) \right\} \tag{2}$$

where there are L number of points (x, y, t) on the trajectory.

In our work, we only consider the motion boundary histogram (MBH) to describe features from the detected trajectories. Unlike the HOF descriptor, MBH uses optical flow information $I_w = (I_x, I_y)$ but computes the spatial derivatives separately for its horizontal (MBHx) and vertical (MBHy) components. These are then used to obtain 8-bin histograms for each component. MBH is also robust to camera and background motions and has reported superior results compared to the HOG and HOF [8]. In detail, the combination of MBHx/MBHy descriptors produces descriptors of size $N \times N \times L$ (N is the size of space-time volume in pixels and L is the length of of trajectories). Each volume is subdivided into $n_x \times n_y \times n_t$ grid of cells; for each cell, 8-bin motion boundary histograms in each direction are then computed. We use the original implementation from [8] and follow standard parameter settings, i.e., $L = 15$, $W = 5$, $N = 32$, $\{n_x, n_y\} = 2$, $n_t = 3$.

3.3 Textural features

We evaluate three types of textural features in our experiments: local binary pattern (LBP), local phase quantization (LPQ), and binarized statistical image features (BSIF). We briefly describe these techniques, followed by how they can be extended for the spatio-temporal case by three orthogonal planes (TOP).

LBP features: Local binary pattern (LBP) [43] uses binary patterns calculated over a region to describe textural properties of an image. The LBP operator describes each image pixel based on the relative gray levels of its neighborhood pixels. If the gray level of the neighboring pixel is higher or equal, the value is set to one, otherwise to zero. The binary pattern is described by considering these binary numbers over the neighborhood as:

$$LBP_{P,R}(x,y) = \sum_{i=0}^{N-1} s(n_i - n_c)2^i, \; s_x = \begin{cases} 1 & x \leq 0 \\ 0 & otherwise \end{cases} \tag{3}$$

where n_c corresponds to the gray level of the center pixel of a local neighborhood, and n_i, the gray levels of N equally spaced pixels on a circle of radius R. The $LBP_{P,R}$ operator produces 2^P possible output values, corresponding to the possible number of binary patterns that can be formed by the P neighborhood pixels. The feature histogram is produced by considering the frequency distribution of the LBP values.

LPQ features: Local phase quantization (LPQ) [44] operator uses local phase information to produce blur-invariant image features extracted by computing the short-term Fourier transform (STFT) in rectangular neighborhoods N_x:

$$F(u,x) = \sum_{y \in N_x} f(x-y)e^{-j2\rho u^T y} = \mathbf{W}_u^T \mathbf{f}_x \tag{4}$$

where W_u is the basis vector of the discrete Fourier transform (DFT) at the frequency u and f_x is a vector that contains all image samples from N_x. Four complex coefficients corresponding to 2D frequencies are considered for forming LPQ features: $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, -a]^T$, where a is a scalar. To express the phase information, a binary coefficient b is formed from the sign of imaginary and real part of these Fourier coefficients. An image is then produced by representing eight binary values (obtained from binary coefficients) as the integer value between 0 to 255. Finally, the LPQ feature histogram is constructed from these output values.

BSIF features: Binarized statistical image features (BSIF) [45] is a more contemporary rmethod that efficiently encodes texture information, in a similar vein to the aforementioned methods that produce binary codes. Given an image X of size $p \times p$, BSIF applies a linear filter F_i that is learnt from natural images by independent component analysis (ICA) [46], on the pixel values of X, obtaining the filter response,

$$r_i = \sum_{u,v} F_i(u,v)X(u,v) = \mathbf{f}_i^T \mathbf{x} \tag{5}$$

where $\mathbf{f}$ and $\mathbf{x}$ are the vectorized form of F_i and X, respectively. The binarized feature b_i is then obtained by thresholding r_i at the level zero, i.e., $b_i = 1$ if $r_i > 0$ and $b_i = 0$ otherwise. The decomposition of the filter mask F_i allows the independent components or basis vectors to be learnt by ICA. Succinctly, we can learn n number of $l \times l$ linear filters W_i, stacked into a matrix $\mathbf{W}$ such that all responses can be efficiently computed by $\mathbf{s} = \mathbf{Wx}$; where, $\mathbf{s}$ is a vector contains r_i responses. Thus, an n-bit binary code is produced for each pixel; all of which builds the feature histogram for the image.

Spatio-temporal extension of textural features: Motivated by the success of recent works related to the recognition of dynamic sequences [10, 12], we consider the three orthogonal planes (TOP) approach to extend the 2D textural operators to cater for videos. Given a video (XYT), the TOP approach extracts the texture descriptors along the XY, XT, and YT orthogonal planes where, the XY plane encodes structural information while XT and YT planes encode space-time transitional information. The histograms of all three planes are concatenated to form the final feature histogram. Generally speaking, the textural histogram given a volumetric space of $X \times Y \times T$ can be defined as:

$$h_j^{plane} = \sum_{p \in P} \mathcal{I}\{b(p) = j\} \tag{6}$$

where, $j \in \{1, \ldots, 2^n\}$, p is a pixel at location (x, y, t) at a particular plane, b is the binarized code, and $\mathcal{I}\{.\}$ a function indicating 1 if true and 0 otherwise. The histogram bins of each plane are then normalized to get a coherent description, $\tilde{\boldsymbol{h}}^{plane} = \left\{\tilde{h}_1^{plane}, \ldots, \tilde{h}_{2^n}^{plane}\right\}$. Finally, we concatenate the histograms of all three planes,

$$\boldsymbol{H} = \left\{\tilde{\boldsymbol{h}}^{XY}, \tilde{\boldsymbol{h}}^{XT}, \tilde{\boldsymbol{h}}^{YT}\right\} \tag{7}$$

In this work, we have set the neighborhood and radius parameters for non-uniform pattern LBP-TOP as $\{P_{XY}, P_{XT}, P_{YT}\} = 8$ and $\{R_X, R_Y, R_T\} = 2$, respectively, following the specifications in [12]. Meanwhile, neighborhood parameters for LPQ-TOP are set to $\{W_x, W_y, W_t\} = 5$, as specified in [47]. For BSIF-TOP, the filter size $l = 9$ and representative bit size $n = 12$ were empirically determined and applied to all three planes.

4 Experimental setup

In this section, we discuss the datasets used and their respective evaluation protocols, as well as details on the implemented experimental pipeline.

4.1 Datasets and their evaluation protocols

In order to exhibit the potency of our proposed methods, we conduct a series of extensive experiments on low quality versions or subsets of three popular benchmark action recognition datasets: the KTH [26], the UCF Youtube [48], and the HMDB [29].

The *KTH* action dataset [26] is one of the most widely used datasets for action recognition. It consists of videos captured from a rather controlled environment, containing 6 action classes performed by 25 actors in 4 different scenarios. There are 599 video samples in total (one subject has one less clip), and each clip is sampled at 25 *fps* at a frame resolution of 160 × 120 pixels. We follow the original experimental setup specified in [26], reporting the average accuracy over all classes. Similar to the protocol established in our previous work [1, 2], six downsampled versions of the KTH were created–three for spatial downsampling (SD_α), and three for temporal downsampling (TD_β). We limit our experiments to downsampling factors, $\alpha, \beta = \{2, 3, 4\}$, which denotes spatial or temporal downsampled versions of a half, a third, and a fourth of the original resolution or frame rate.

These videos that undergo spatial downsampling lose many important spatial details which may fail interest point detectors. To cope with this issue, we increase the sensitivity of the change of gradients to detect a decent amount of interest points from each videos. Specifically, we use various k parameters for different downsampled modes, i.e., $k = 0.0001, 0.000075$, and 0.00005 for SD_2, SD_3, and SD_4 respectively. Since there is no change in frame resolution for the case of temporal downsampling so, we keep the value of k parameter unchanged. Also for estimation of feature trajectories, we also use different values for neighborhood size N and trajectory length L, i.e., we use $N = 11$, 8, and 4 for SD_2, SD_3, and SD_4 videos respectively and $L = 15$, 8, and 5 for TD_2, TD_3, and TD_4 videos, respectively. We empirically determine the suitability of these values by prior experiments.

The *UCF-YouTube* [48], also known 'UCF-11' is another popular dataset for action recognition, consisting of videos captured from uncontrolled and complex environments. It contains 11 action classes, and every class has 25 groups with more than 4 action clips in each group. The video clips that belong to the same group share some common features, such as the same actor, similar background, and similar viewpoint. The videos are compromised with various problems such as camera motion, background clutter, viewpoint, and scale variations. There are 1600 video samples in total and each clip is sampled at ∼30 fps with a frame resolution of 320 × 240 pixels. We follow the leave-one-group-out-cross-validation (LOGOCV) scheme specified in [48], reporting the average accuracy over all groups. Since we are interested in evaluating low-quality videos, we apply lossy compression on each video sample. Specifically, we re-encode all video samples by using ×264 video encoder [49] by randomly assigning constant rate factors (crf) that are uniformly distributed across all samples. We used crf values between 23 to 50 where higher values indicate greater compression (and smaller file sizes) and vice versa. For clarity, we call this newly version, *YouTube-LQ*, with videos now of low quality due to the effects of lossy compression. Some sample videos created with different crf values are shown in Fig. 3. As we can see from the figure, videos that has a higher crf values have poor structural information.

The *HMDB* [29] is one of the largest human action recognition dataset that is increasingly popular in recent years. It has a total of 6766 videos of 51 action classes collected from movies and YouTube videos. HMDB is a considerably challenging dataset with videos acquired from uncontrolled environments with large viewpoint, scale, background, and illumination variations. Videos in HMDB are annotated with a rich set of meta-labels including quality information: three quality labels were used, i.e., "good," "medium," and "bad." Three training-testing splits were defined for the purpose of evaluations, and performance is to be reported by the average accuracy over all three splits. In our experiments, we use the same specified splits for training, while testing was done using only videos with "bad" and "medium" labels; for clarity, these two sets will hereafter be denoted as *HMDB-BQ* and *HMDB-MQ*, respectively. In the medium quality videos, only large body parts are identifiable, while they are totally unidentifiable in the bad quality videos due to the presence of motion blur and compression artifacts. Bad and medium videos comprise of 20.8 and 62.1% of the total number of videos in the entire original database respectively.

Figure 1 shows some sample frames of various actions from the downsampled KTH dataset, compressed YouTube dataset and "poor" quality subset of the HMDB subset.

4.2 Evaluation framework setup

Our evaluation framework generally comprises of two main steps: feature representation and classification.

For feature representation, spatio-temporal features are first extracted from each action video before encoding into a "histogram of visual words" using visual codewords generated by classic bag-of-visual-words (BoVW) method. In all our experiments, we perform histogram-level concatenation of two types of features: encoded HOG and HOF descriptors for interest point method (denoted by "STIP") and encoded MBHx and MBHy descriptors for the trajectory-based method (denoted by 'iDT'). Histogram-level concatenation is known to be more effective than descriptor-level concatenation [1, 2]. Feature histograms from various dynamic textural features, such as LBP-TOP, LPQ-TOP, and BSIF-TOP, are also extracted from the videos, and then concatenated with their associated encoded features. In our experiments, we set the codebook size to 4000 which has been empirically shown to be effective in obtaining good results across numerous datasets [6]. To decrease the computational overhead during codebook generation, we used a subset of 100,000 features randomly selected from all training samples.

To perform a classification, we use a multi-class non-linear support vector machine (SVM) with χ^2-kernel defined as:

$$K(H_i, H_j) = exp(-\gamma D(H_i, H_i)) \tag{8}$$

where $H_i = \{h_{i1}, \ldots, h_{in}\}$ and $H_j = \{h_{j1}, \ldots, h_{jn}\}$ denote histograms of visual words. D is the χ^2 distance function defined as:

$$D(H_i, H_j) = \sum_{i=1}^{V} \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}} \tag{9}$$

where V is the size of codebook and γ is the mean of distances between all training samples. We use a computationally efficient approximation of the non-linear kernel by Veldadi and Zisserman [50] which allows features to undergo a non-linear kernel map expansion before SVM classification. It provides us the flexibility of deciding which features are to be "kernelized." We fixed the value of regularization parameter c to 10 and adopted a *one-versus-rest* strategy for multi-class classification, where classes with the highest score are considered as the predicted class.

For benchmarking, we regard "STIP" and "iDT" features as our baseline methods and also made comparisons against other competing methods from literature.

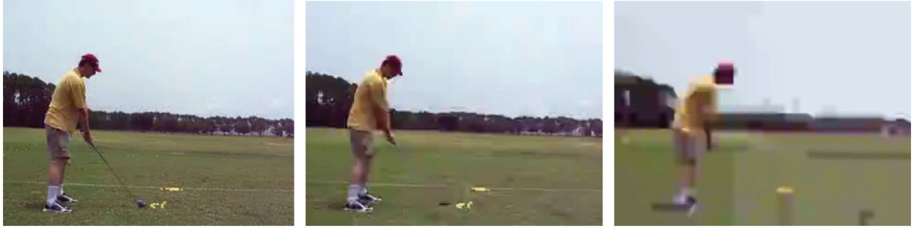

Fig. 3 Sample video frame with different constant rate factors (crf). Left: crf 29, center: crf 38, right: crf 50. Note the adverse deterioration in quality when video is compressed

5 Results and analysis

In this section, we present the results of a comprehensive set of experiments conducted on the three datasets, followed by an in-depth analysis into the results and various influencing factors.

5.1 Experiments on downsampled KTH videos

In this section, we present our experimental results on six downsampled versions of the KTH dataset. We choose the KTH dataset to perform this extensive range of experiments as it is lightweight and also a widely used benchmark in this domain area. From the results reported in Table 1 (STIP-based methods) and Table 2 (iDT-based methods), methods that exploit additional textural features clearly demonstrate significant improvement, as compared to their respective baseline methods. This is notably consistent across all six downsampled videos of KTH dataset. Also, methods that used iDT as the base feature outperform their STIP-based counterparts across all downsampled versions. Among the textural features employed, BSIF-TOP appears to be the most promising choice, clearly outperforming the other textural features. More important, it must be pointed out that the contribution of textural features becomes more significant as video quality deteriorates (particularly for cases SD_4 and TD_4). This exemplifies the robustness of textural features against video quality.

Analysis on experiments: From the results of both STIP and iDT features, it is observed that the decrease in performance is most obvious with respect to spatial resolution. Feature detection in image frames is based on the variation of intensities in local image structures. Hence, the drop in spatial resolution may cause failure in detecting essential image structures. Compared to STIP, the performance of iDT features dropped tremendously, especially for SD_3 and SD_4 videos. Figure 4 gives a closer look on the detected features when videos are downsampled spatially and temporally. While the number of detected iDT features appear to be more than STIP features, they are obviously less salient (Fig. 4 shows a lot of trajectories that were sampled from the background regions).

Table 1 Recognition accuracy (%) of various STIP-based approaches in comparison with other approaches on downsampled versions of the KTH dataset

Method	SD_2	SD_3	SD_4	TD_2	TD_3	TD_4
STIP (baseline)	86.85	80.37	75.56	88.24	82.31	78.98
STIP+LBP-TOP	85.19	82.04	77.59	88.43	82.41	81.20
STIP+LPQ-TOP	87.41	80.19	76.30	87.41	81.85	79.81
STIP+BSIF-TOP	88.80	85.28	81.67	88.70	86.11	84.54
STIP+Deep object [35]	82.41	82.41	81.48	82.41	80.56	80.09
LTTS+DS model [34]	82.22	83.78	80.00	–	–	–

Table 2 Recognition accuracy (%) of various iDT based approaches in comparison with other approaches on downsampled versions of the KTH dataset

Method	SD_2	SD_3	SD_4	TD_2	TD_3	TD_4
iDT (Baseline)	92.59	78.80	61.85	95.19	91.57	89.54
iDT+LBP-TOP	92.96	81.94	73.61	95.09	92.13	89.54
iDT+LPQ-TOP	92.96	78.61	79.91	95.09	91.67	88.89
iDT+BSIF-TOP	93.89	88.33	82.41	95.09	92.22	90.00
iDT+Deep object [35]	86.57	84.26	82.41	87.04	85.19	84.26
LTTS+DS model [34]	82.22	83.78	80.00	–	–	–

Spatio-temporal textures circumvent this feature detection step by relying on statistical regularities across the spatio-temporal cube. Regions in an image such as background areas that have less textural information will offer little count towards the overall statistics. However, previous findings [2, 14] have observed that textural features alone do not offer good performance though it can serve as a strong supplement to other attention-oriented features such as shape and motion. For instance, in case of STIP based methods, BSIF-TOP textures help improve the accuracy for both SD_4 and TD_4 videos by $\approx 6\%$; for iDT based methods, it improves by $\approx 21\%$ for SD_4 and $\approx 0.5\%$ for TD_4 videos.

Among various textural features used jointly with STIP and iDT features, BSIF-TOP appears to be the most promising choice, as it outperforms the rest. With the degradation of spatial resolution and temporal sampling rate, BSIF-TOP comparatively performs better than LBP-TOP and LPQ-TOP features. Figures 5 and 6 analyzes the the performance improvement of BSIF-TOP features relative to that of LBP-TOP and LPQ-TOP. For instance, on STIP features, the improvement of BSIF-TOP over LPQ-TOP is $\approx 5\%$ for SD_4 and $\approx 4.8\%$ TD_4 videos. On iDT features, the improvement of BSIF-TOP over LPQ-TOP is $\approx 10\%$ for SD_3 and $\approx 0.5\%$ for TD_3 videos. Overall observation points to the fact the BSIF-TOP performs relatively well as the video quality drops, an indication of its robustness in this aspect.

Our best approach, which combines the base features with BSIF-TOP dynamic textures, also performed better than the recent works by Gao et al. [34] and Rahman and See [35]. Surprisingly, the results of [34] are reported without the "*s2*" videos from KTH, which are videos that contain larger motion and scale variations. This could suggest that their method could fare worser still with the consideration of the omitted "*s2*" videos.

We further furnish the confusion matrices for four approaches: STIP, STIP+BSIF-TOP, iDT, and iDT+BSIF-TOP on the SD_3 videos in Fig. 7. Due to space limitations,

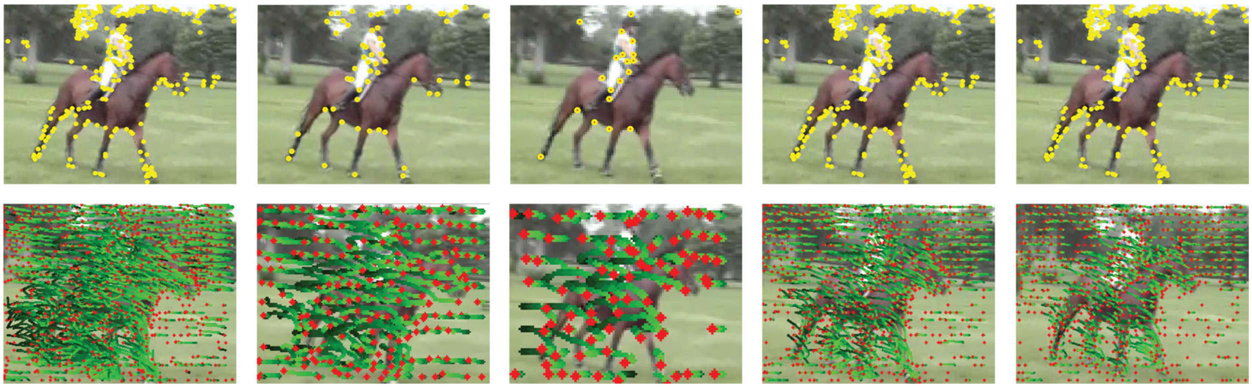

Fig. 4 Response of detectors when videos are downsampled spatially and temporally. Videos on the first row are using Harris3D detector and videos on second row are using warped flow estimation based feature tracking. The videos in columns 1, 2, and 3 represents the baseline, half resolution of the baseline, and one-third resolution of the baseline, respectively, while columns 4 and 5 represents one-half and one-third frame rate of baseline, respectively

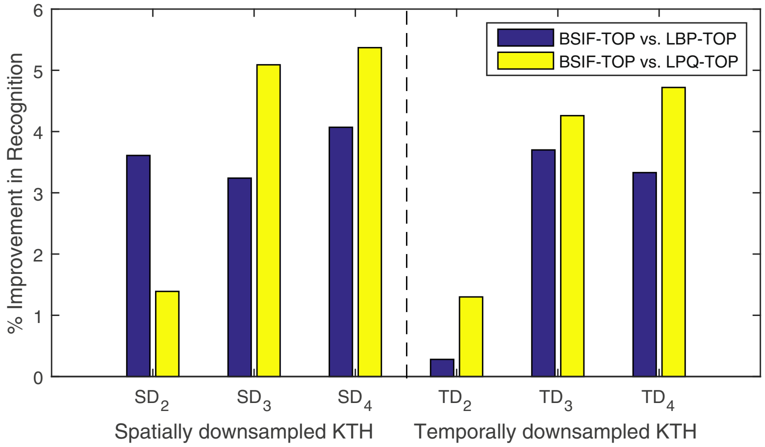

Fig. 5 Percentage improvement of BSIF-TOP over LBP-TOP and LPQ-TOP when combined with STIP on downsampled KTH

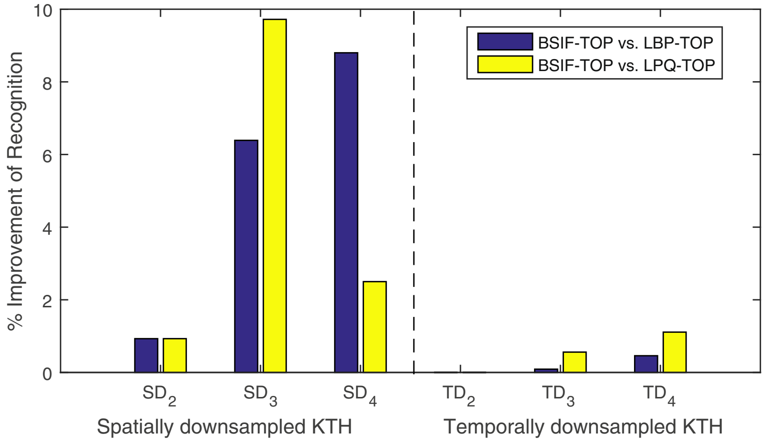

Fig. 6 Percentage improvement of BSIF-TOP over LBP-TOP and LPQ-TOP, when combined with iDT on downsampled KTH

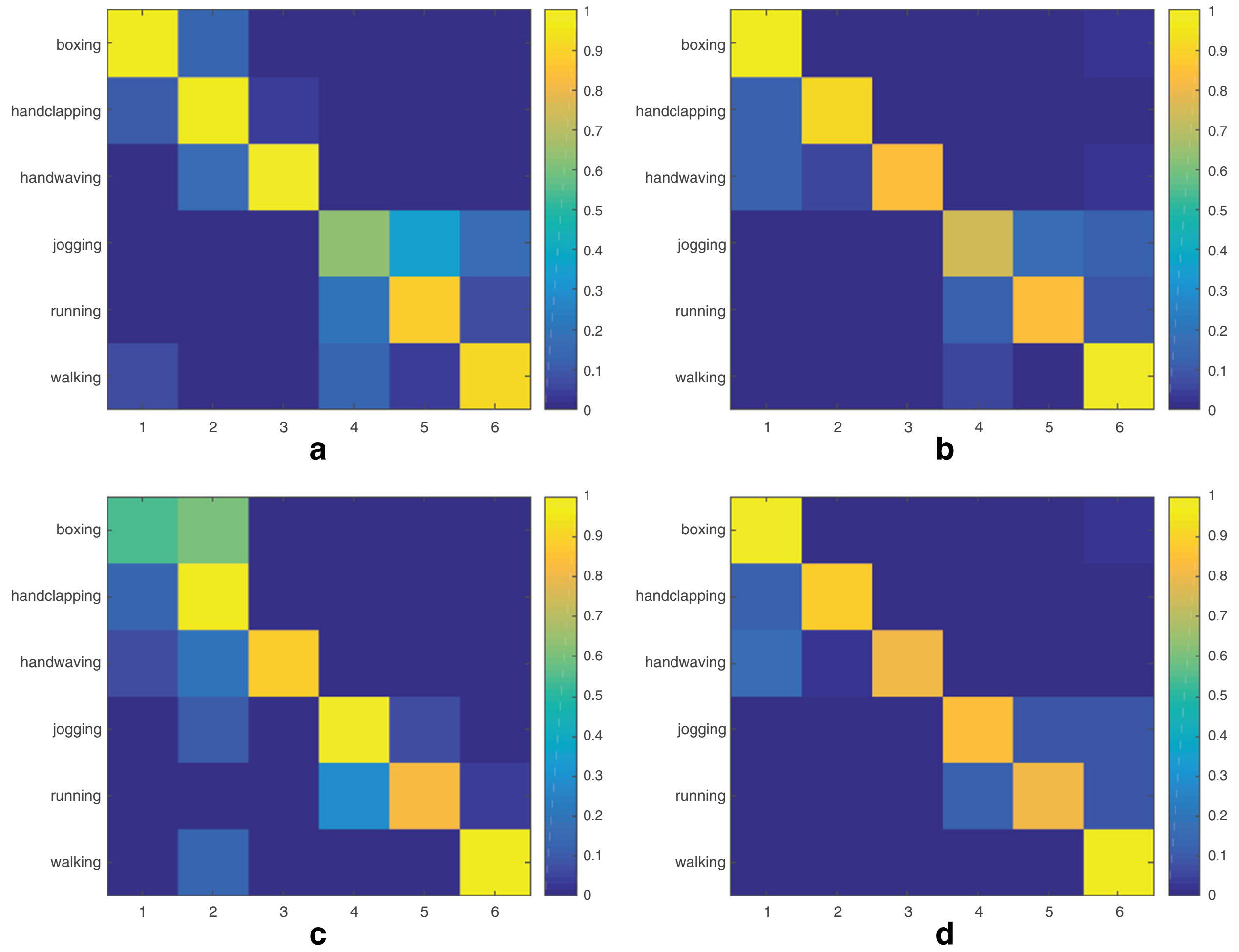

Fig. 7 Confusion matrices of the KTH-SD_3 dataset. Confusion matrices on the right side show the effects of fusing BSIF-TOP textural features with STIP and iDT features. **a** STIP. **b** STIP+BSIF-TOP. **c** iDT. **d** iDT+BSIF-TOP

we only report these confusion matrices for the SD_3 videos as example. For both STIP and iDT features, it is clear to see that additional usage of textural features helps to improve the accuracy of certain action classes such as“walking” and “jogging” by more than ≈20–40%. However, this is also at the expense of a slight drop in accuracy for other classes such as “handclapping” and “handwaving.” We make a special note for the recognition of “boxing” videos using feature trajectories: Its somewhat poor performance on low resolution videos is greatly improved through the introduction of textural features, an increase of more than 50%.

5.2 Experiments on compressed videos of YouTube dataset (YouTube-LQ)

In this section, we repeated our experiments on on compressed videos of YouTube dataset (YouTube-LQ) to demonstrate the effectiveness of using textural features with both STIP and iDT features. We observe that after applying compression, both the STIP and iDT baseline features struggle to maintain their original performances (i.e., 71.94% for STIP and 81.58% for iDT). From Table 3, it is clear that methods that use additional textural features demonstrate significant improvement as compared to their baselines. Again, iDT-based methods outperform their STIP counterparts while joint usage with BSIF-TOP once again tops the other textural features by a good measure. However, our best textural feature still falls short compared to the use of deep object features [35], which is arguably very robust against effects stemming from video compression. We endeavor to investigate in future how textural and deep object features can be synergized together.

Analysis on experiments: After applying compression, the performance of baseline features become lower than that on the original videos since it critically affects the spatial quality. With the inclusion of textural features, the

Table 3 Recognition accuracy (%) of various approaches on the Youtube-LQ dataset

Method	YouTube-LQ	Method	YouTube-LQ
STIP (baseline)	67.57	iDT (Baseline)	74.04
STIP+LBP-TOP	70.69	iDT+LBP-TOP	75.59
STIP+LPQ-TOP	69.13	iDT+LPQ-TOP	76.02
STIP+BSIF-TOP	76.05	iDT+BSIF-TOP	80.45
STIP+deep object [35]	85.37	iDT+deep object [35]	85.36

performance of both STIP and iDT features increased significantly. Once again, BSIF-TOP is the most promising choice, offering the highest performance improvement in comparison to LBP-TOP and LPQ-TOP. Figure 8 provides a closer look into how BSIF-TOP improves the recognition performance at a much larger extent over other textural features.

The confusion matrices shown in Fig. 9 offers more insightful analysis into class-wise performances. It is clear that for both STIP- and iDT-based methods, the addition of textural features play an important role. Many action classes have improved, i.e., “golf swing,” “soccer juggling,” “swing,” “tennis swing,” “trampoline jumping” and “volleyball spiking.” It is interesting to mention that the iDT features performed slightly better on actions with complex scenes such as “volleyball spiking” than the STIP features, when combined with BSIF-TOP features.

5.3 Experiments on medium and bad quality subsets from HMDB dataset (HMDB-MQ and HMDB-LQ)

In order to demonstrate the effectiveness of adding texture information to STIP- and iDT-based features for a larger number of action classes, we also run our experiments on low-quality subsets of HMDB dataset. From Table 4, we can see a significant leap in performance when the textural features are aggregated, particularly BSIF-TOP. The iDT+BSIF-TOP method achieved a very commendable result with $\approx 14\%$ improvement on the “Bad” subset (HMDB-LQ) and $\approx 4.5\%$ increase on the “Medium” subset (HMDB-MQ). The method that incorporates deep object features [35] remains competitive against our proposed methods. It appears to perform better when STIP is the choice of base feature. Nevertheless, our proposed usage of textural features, particularly the BSIF-TOP, still surpasses the performance of the method with deep object features when iDT is the base feature.

Analysis on experiments: On both STIP and iDT base features, the “Bad” quality videos are the most challenging, with only a recognition accuracy of just above 20%. The use of textural features are able to help increase their performances by a very good margin of ≈ 11 and $\approx 14\%$ for STIP and iDT features, respectively. Meanwhile, for “Medium” quality videos, the improvement in performance after supplementing with textural features is not as marked on the iDT base features as compared to the STIP base features. The addition of BSIF-TOP features offers the most significant jump in performance, almost 9% more than the next best textural feature (LPQ-TOP). Further analysis on the performance improvement of the BSIF-TOP over its textural counterparts is shown in Fig. 10. As expected, the BSIF-TOP is far more superior than the LBP-TOP particularly when STIP features are considered as the base features. These differences are much less pronounced when combined with iDT features instead.

Figure 11 show the confusion matrices for the STIP, STIP+BSIF-TOP, iDT, and iDT+BSIF-TOP features respectively, based on the first split of the HMDB dataset (both HMDB-LQ and HMDB-MQ included). Since there are 51 classes in total, we can only compare them visually by observing the diagonal patterns in these confusion matrices. The more coloured the diagonals are on a blue background, the better the performance with lesser false positives. In total, about 15–17 action classes improved after BSIF-TOP is used together with the base features. Some action classes that benefit from the use of textural information are such as “Climb,” “Sword,” “Draw sword,” “Fencing,” and “Golf.”

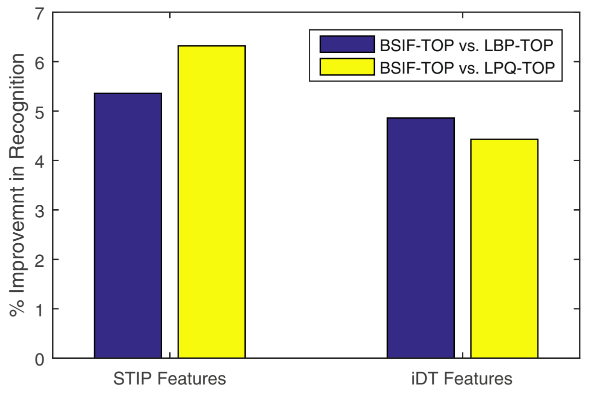

Fig. 8 Percentage improvement of BSIF-TOP over LBP-TOP and LPQ-TOP, when combined with trajectory based features on YouTube-LQ

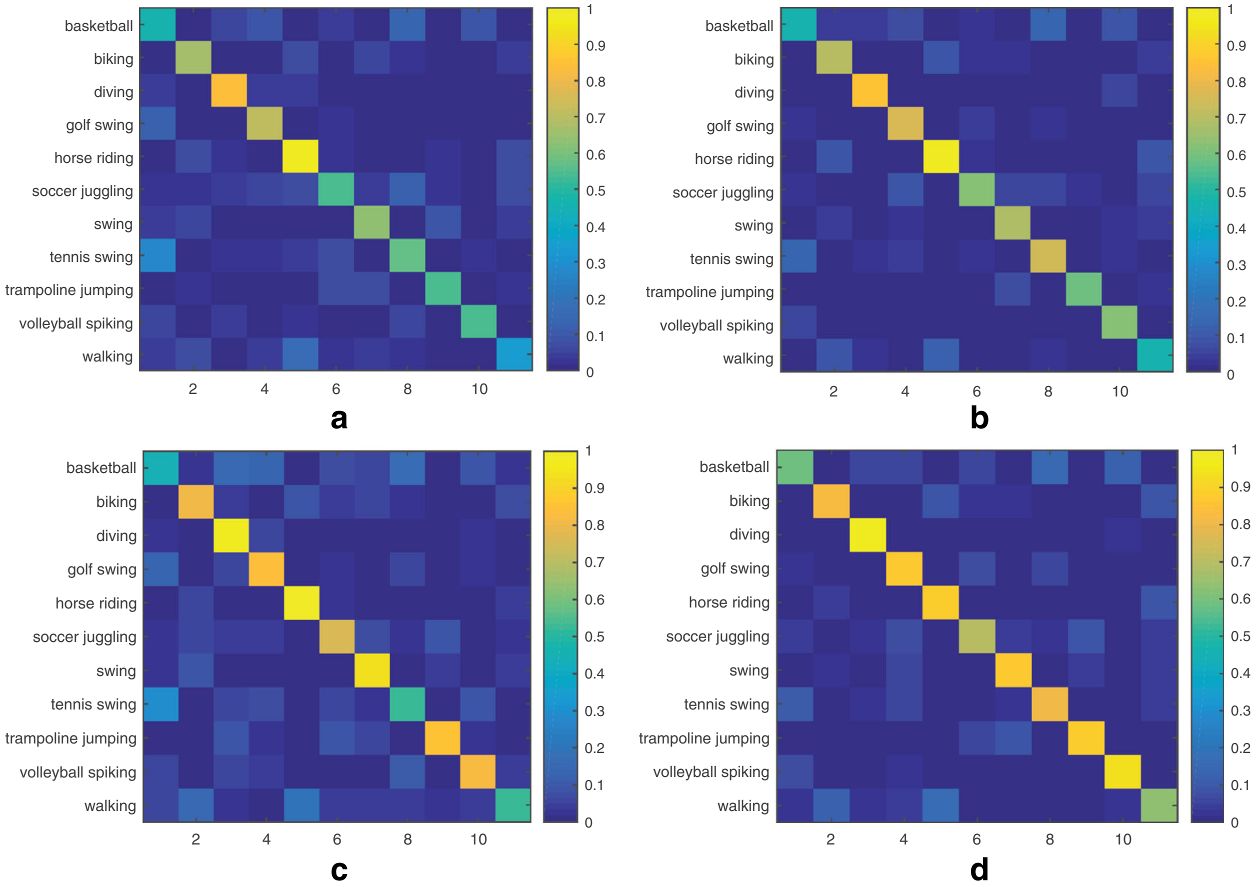

Fig. 9 Confusion matrices of the YouTube-LQ dataset. Confusion matrices on the right side show the effects of fusing BSIF-TOP textural features with STIP and iDT features. **a** STIP. **b** STIP+BSIF-TOP. **c** iDT. **d** iDT+BSIF-TOP

5.4 Discussion

In this section, we present additional analysis as a result of our investigation into several influencing factors in our recognition pipeline such as the comparison between the textural features considered in this work, feature sampling for codebook generation and the choice of feature encoding method. We also offer a balanced commentary on the potential use of deeply learned features.

Analysis on textural features: To investigate the efficacy of textural features alone, we remove the base shape and motion features (STIP and iDT) for the purpose of this analysis. Figure 12 compares the performance of the three dynamic textural features considered in this paper: LBP-TOP, LPQ-TOP, and BSIF-TOP. This was done for the original and six downsampled versions of the KTH dataset (denoted as SD_2, SD_3, SD_4, TD_2, TD_3, TD_4), the original and compressed UCF-YouTube datasets, and the three subsets of the HMDB dataset (denoted as "Bad," "Medium," "Good"). In all cases, the BSIF-TOP emerged as the most robust textural feature, capable of extracting effective global information regardless of the adversity in video quality.

Table 4 Recognition accuracy (%) of various feature combinations on HMDB-LQ and HMDB-MQ subsets

Method	HMDB-LQ	HMDB-MQ	Method	HMDB-LQ	HMDB-MQ
STIP (baseline)	21.71	23.68	iDT (Baseline)	23.88	41.43
STIP+LBP-TOP	20.80	24.28	iDT+LBP-TOP	30.34	43.11
STIP+LPQ-TOP	23.89	28.36	iDT+LPQ-TOP	30.96	42.97
STIP+BSIF-TOP	32.46	37.14	iDT+BSIF-TOP	37.80	45.96
STIP+deep object [35]	34.57	42.48	iDT+deep object [35]	36.51	44.80

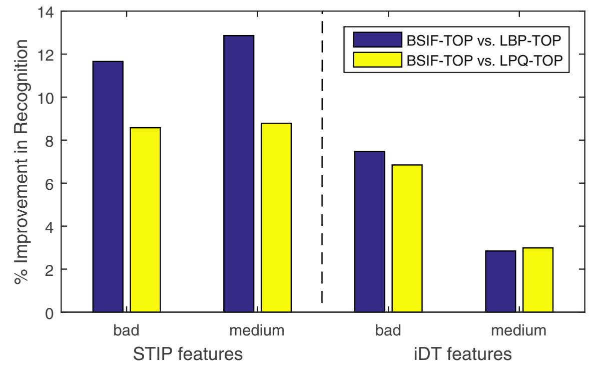

Fig. 10 Percentage improvement of BSIF-TOP over LBP-TOP and LPQ-TOP, when combined with interest point based features on HMDB-LQ and HMDB-MQ

Analysis on feature sampling for codebook generation: Determining the appropriate codebook size is important to ensure that the extracted local features are encoded into a codebook of sufficient capacity. Many authors have analyzed this issue in detail [6, 37] and have also suggested appropriate codebook sizes based on their empirical evaluations over many experimental datasets. Following their suggestions, we choose to use a codebook size of 4000 for all our tested datasets, after verification by experiments. However, the number of features that are sampled randomly to build the codebook could potentially be vital to the recognition performance. For consistency in our main experiments (in Sections 5.1, 5.2, and 5.3), we had fix the number of sampled features to 100,000 to obtain a reasonably good level of accuracy while maintaining a manageable computational load for codebook learning.

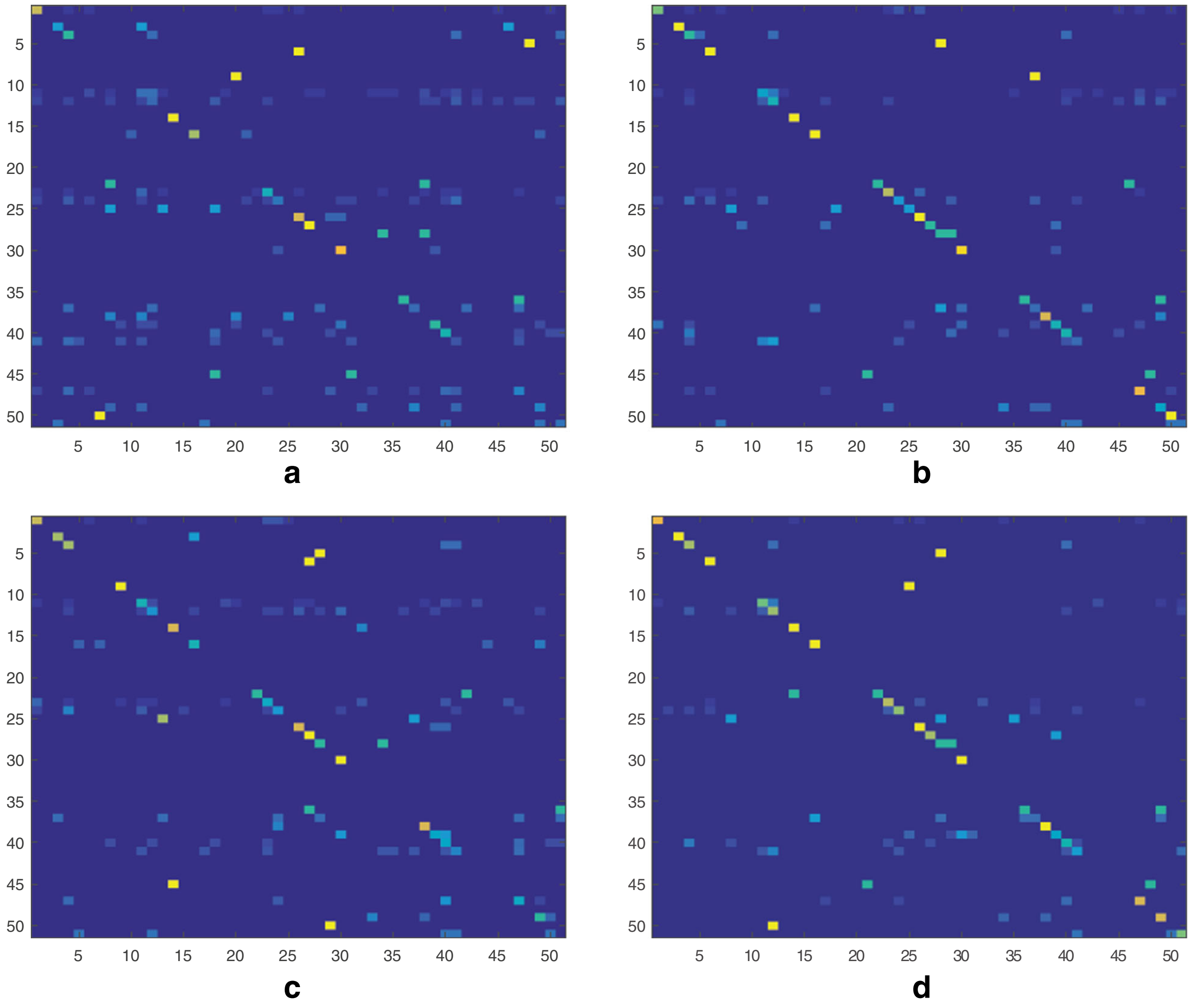

Fig. 11 Confusion matrices of the first split of HMDB dataset. Confusion matrices on the right side show the effects of fusing BSIF-TOP textural features with STIP and iDT features. **a** STIP. **b** STIP+BSIF-TOP. **c** iDT. **d** iDT+BSIF-TOP

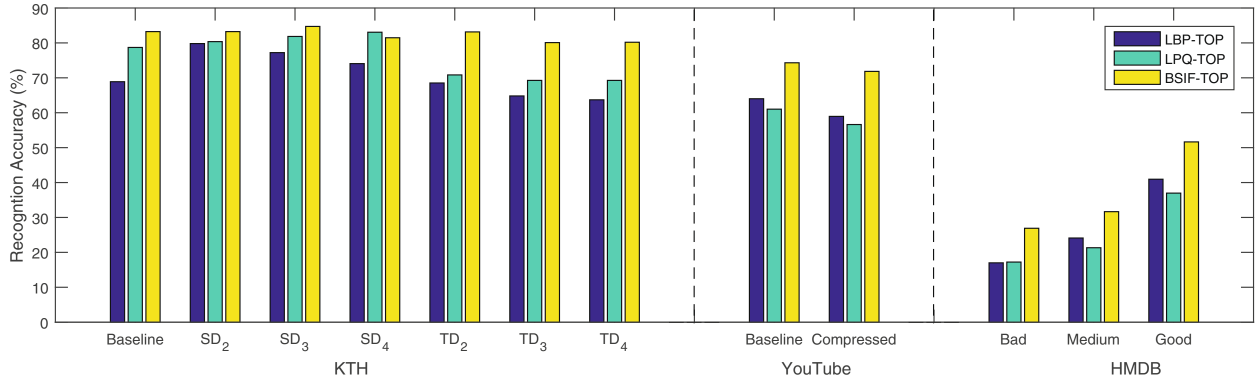

Fig. 12 Performance of various textural features on the original and low quality datasets

To see the effect of using more features for codebook generation, we tested two of our best performing methods on all three subsets of the HMDB dataset, with a variety of feature set size: 100, 150, 200, and 250 k. In Fig. 13, we observe that the recognition accuracy of various HMDB videos somewhat improves when we use more features to construct the codebook (codebook size remains the same at 4000). Interestingly, with the STIP as base feature (see Fig. 13a), we can achieve better accuracy of up to ≈5%) if we use a larger sampling size. But the scenario is the opposite for the iDT case (see Fig. 13b) where the recognition accuracy significantly drops across all three subsets when larger sampling sizes are used. The iDT features are constructed by MBHx and MBHy descriptors, which describe the "gradient" on the temporal dimension of the trajectories, along both horizontal and vertical spatial directions. Hence, if too many features are sampled, the intra-class variations of these trajectories may likely result in a variety of perturbations to the descriptors, which in turn increases the ambiguity during clustering. A possible remedy to this is to increase the codebook size to accommodate a larger variety of trajectories, particularly for more complex scenes such as those in HMDB. Codebooks constructed from less ambiguous features have higher discriminative capacities that may help to gain better recognition performance.

Analysis on shape and motion features encoding Many feature encoding methods have been proposed in literature, such as histogram encoding, Fisher Vector encoding and sparse coding [37]. We choose to use histogram encoding, better known as bag-of-visual-words (BoVW) since it is widely used by many recent action recognition works [2, 6, 7]. Though some authors [1, 8]

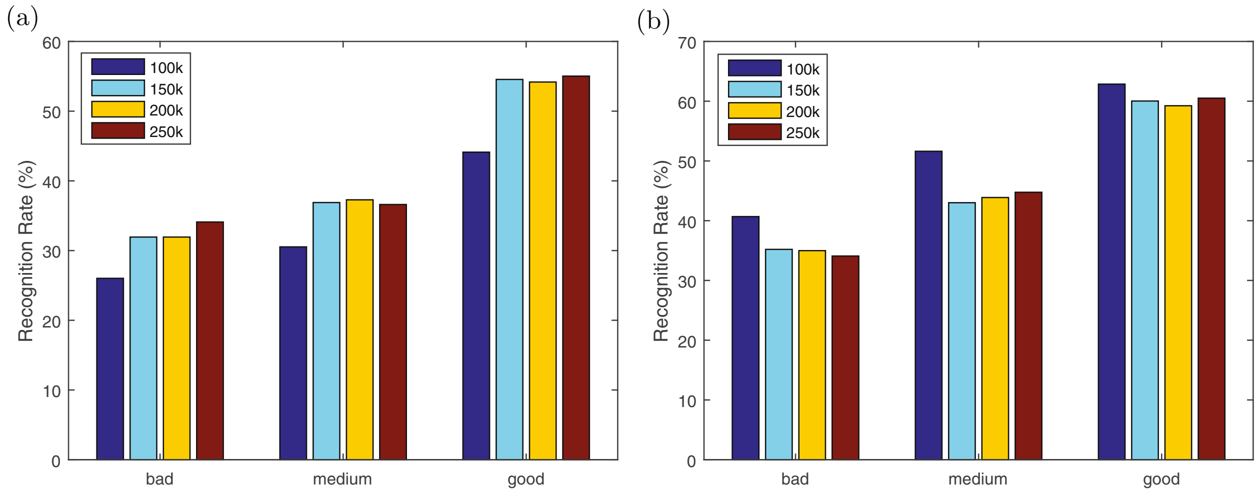

Fig. 13 Recognition performance of STIP+BSIF-TOP and iDT+BSIF-TOP on various subsets of the HMDB dataset. Bars of different colors denote the varying amount of feature descriptors (100, 150, 200, 250 k) that are sampled for codebook generation. **a** STIP+BSIF-TOP. **b** iDT+BSIF-TOP

showed that Fisher Vector (FV) encoding is superior to BoVW encoding, but we found that this is untrue most of the time for the evaluated low-quality videos, with the exception of videos from HMDB. Our best methods (STIP+BSIF-TOP and iDT+BSIF-TOP) achieve better accuracy using BoVW encoding on most counts, as can be seen in Table 5. From this observation, we suggest that codebooks for videos with plenty of motion and complex background scenes may be better constructed using FV encoding which applies soft quantization to features.

Analysis on deeply learned features: Owing to the recent breakthrough in deep learning techniques, particularly deep convolutional neural networks (CNNs), we have made comparisons against a recent work that used object features extracted from a pre-trained CNN model [35]. In this work, the authors used an popular off-the-shelf CNN model, the VGG-16 architecture [51] that was pre-trained on ImageNet for 1000 categories. Object features from the last few layers of the network (concatenation of *fc6* and *fc7* layers) were extracted from each frame and average pooled.

Interestingly, the widely acclaimed deeply learned features was not entirely superior in all cases. Experimental results on the downsampled KTH datasets (Tables 1 and 2) and the HMDB low-quality subsets (Table 4) showed that the combination of the robust BSIF-TOP dynamic textural feature with the base features (STIP or iDT) can surpass the recognition capability of combining with deeply learned object features. In fact, the baseline performance for some of the downsampled versions (particularly SD2, TD2, TD3) is also better than that combined with deep object features. This can be attributed to the pre-trained CNN model's inability to generalize for videos with distorted spatial information. To obtain a single dimension, the average-pooled deep features (across all frames) [35] also resulted in the removal of valuable temporal cues.

Table 5 Recognition accuracy (%) of various datasets with STIP+BSIF-TOP and iDT+BSIF-TOP methods using bag-of-visual-words (BoVW) and fisher vector (FV) encoding

	STIP+BSIF-TOP		iDT+BSIF-TOP	
Datasets	BoVW	FV	BoVW	FV
KTH-SD_2	88.80	89.26	93.89	92.87
KTH-SD_3	85.28	83.15	88.33	87.78
KTH-SD_4	81.67	80.19	82.41	81.02
KTH-TD_2	88.70	89.91	95.09	94.44
KTH-TD_3	86.11	87.78	92.22	92.59
KTH-TD_4	84.54	82.96	90.00	90.28
Youtube-LQ	76.05	75.04	80.45	78.13
HMDB-BQ	32.46	33.06	37.80	40.69
HMDB-MQ	37.14	38.51	45.96	51.62

However, for the compressed YouTube-LQ dataset, the use of deep object features performed exceedingly better than our best approach (see Table 3). The BSIF-TOP feature dimension is also larger than that of the *fc6*+*fc7* deep object features (12,288 to 8192); this will cause classifier training to take a longer time.

5.5 Computational complexity

In this section, we compare the various feature descriptors (including time consumed for their feature detection) based on their total computation speed. This comparison is performed using a sample “Bike riding” video taken from the HMDB dataset, with a 240×320 frame resolution and a total of 246 image frames at 30 *fps*. This estimation of run-time speed was performed on an Intel i7 3.60 GHz machine with 24 GB RAM. Table 6 shows the computational cost of various feature descriptors in seconds, per image frame. Among the shape-motion descriptors, the HOG+HOF feature takes 0.047 s faster than the MBHx+MBHy feature, which relies on feature tracking and warped flow estimation. Among the textural features compared, the LPQ-TOP and BSIF-TOP are the most efficient methods (both much faster than computing shape-motion features), and yet they are also the most promising features for recognizing actions in low-quality video. Between the two feature detectors, extracting the iDT (the better performing method) takes around 0.2 s per frame on a single scale and much longer on multiple scales. In future, we intend to explore the possibility of multi-scale trajectories with the help of parallelized frameworks [52].

6 Conclusion

In this paper, we demonstrate that dynamic textural features can help improve the performance of action recognition in low-quality videos by a good margin. In comparison with current methods that mainly rely on shape and motion features, the use of textural features is a novel proposition that is found to be robust against undesirable, but often, realistic video conditions: low resolution and frame rate, lossy compression, and the presence of motion blurring and artifacts. Our extensive set of experiments marked the BSIF-TOP as a promising candidate for textural features to complement conventional shape and motion features.

Table 6 Computational cost (detection+description) of various feature descriptors

	STIP	iDT	LBP-TOP	LPQ-TOP	BSIF-TOP
Time per frame (in sec.)	0.156	0.203	1.230	0.041	0.051

Even with the advent of deep learning techniques, we see a great value in the use of features that directly exemplify a particular image structure such as textures. However, features learned from deep neural networks have also showed great potential, even more so if the network has been carefully tuned for the target domain. Likewise, the filters used in the BSIF approach are fundamentally learnt through ICA in an unsupervised manner. Hence, future directions point towards further exploration on how richer features can be learnt from videos sampled from a wide quality range to enable better generalization.

Acknowledgements
The authors thank the Ministry of Higher Education, Malaysia and Multimedia University, Malaysia for supporting this research. They also wish to thank the anonymous reviewers for their comments and suggestions which helped in enhancing the quality of the paper.

Funding
This work was supported, in part, by the Ministry of Higher Education, Malaysia under Fundamental Research Grant Scheme (FRGS) project FRGS/2/2013/ICT07/MMU/03/4 and Multimedia University, Malaysia.

Availability of data and materials
The constant rate factors (crf) that were used to compress all 1600 videos of the UCF-YouTube dataset to construct the YouTube-LQ dataset are made publicly available at: http://saimunur.github.io/YouTube-LQ-CRFs.txt . These crf values are randomly assigned to the videos according to uniform distribution. We use crf values between 23–50; higher values indicate greater compression and vice versa.

Authors' contributions
SR and JS contributed equally to this work. SR implemented the research idea, constructed the low quality datasets, and performed the evaluation. JS carried out the in-depth analysis of the experimental results and checked the correctness of the algorithms and evaluation. All authors took part in the writing and proof reading of the final version of the paper. All authors read and approved the final manuscript.

Authors' information
All authors are affiliated to the Centre of Visual Computing, Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia. SR is currently at ViTrox Corporation Berhad, Penang, Malaysia.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 6 June 2017 Accepted: 25 October 2017
Published online: 21 November 2017

References

1. J See, S Rahman, in *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference On*. On the effects of low video quality in human action recognition (IEEE, 2015), pp. 1–8
2. S Rahman, J See, CC Ho, in *IEEE Int. Conf. on Signal and Image Processing Applications (ICSIPA)*. Action recognition in low quality videos by jointly using shape, motion and texture features (IEEE, 2015), pp. 83–88
3. JJ Donovan, D Hussain, *Video data storage, search, and retrieval using meta-data and attribute data in a video surveillance system*. (Google Patents, 2008). US Patent 7,460,149
4. G Gualdi, A Prati, R Cucchiara, Video streaming for mobile video surveillance. IEEE Trans. Multimed. **10**(6), 1142–1154 (2008)
5. I Laptev, On space-time interest points. Int. J. Comput. Vis. **64**(2-3), 107–123 (2005)
6. H Wang, MM Ullah, A Klaser, I Laptev, C Schmid, in *BMVC 2009-British Machine Vision Conference*. Evaluation of local spatio-temporal features for action recognition (BMVA Press, 2009), pp. 124–1
7. H Wang, A Kläser, C Schmid, C-L Liu, in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On*. Action recognition by dense trajectories (IEEE, 2011), pp. 3169–3176
8. H Wang, C Schmid, in *Computer Vision (ICCV), 2013 IEEE International Conference On*. Action recognition with improved trajectories (IEEE, 2013), pp. 3551–3558
9. N Dalal, B Triggs, in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference On*. Histograms of oriented gradients for human detection, vol. 1 (IEEE, 2005), pp. 886–893
10. G Zhao, M Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern. Anal. Mach. Intell. **29**(6), 915–928 (2007)
11. V Kellokumpu, Z Guoying, M Pietikäinen, in *BMVC*. Human activity recognition using a dynamic texture based method, vol. 1 (BMVA Press, 2008), p. 2
12. R Mattivi, L Shao, in *Computer Analysis of Images and Patterns*. Human action recognition using lbp-top as sparse spatio-temporal feature descriptor (Springer, 2009), pp. 740–747
13. SMM Ahsan, JK Tan, H Kim, S Ishikawa, in *Image Processing (ICIP), 2014 IEEE International Conference On*. Histogram of dmhi and lbp images to represent human actions (IEEE, 2014), pp. 1440–1444
14. H Kataoka, Y Aoki, K Iwata, Y Satoh, in *Visual Computing (ISVC), 11th International Symposium On*. Evaluation of vision-based human activity recognition in dense trajectory framework (Springer, 2015), pp. 634–646
15. KK Reddy, N Cuntoor, A Perera, A Hoogs, in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference On*. Human action recognition in large-scale datasets using histogram of spatiotemporal gradients (IEEE, 2012), pp. 106–111
16. F Harjanto, Z Wang, S Lu, AC Tsoi, DD Feng, Investigating the impact of frame rate towards robust human action recognition. Sign. Process. **124**, 220–232 (2016)
17. P Dollár, V Rabaud, G Cottrell, S Belongie, in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop On*. Behavior recognition via sparse spatio-temporal features (IEEE, 2005), pp. 65–72
18. G Willems, T Tuytelaars, L Van Gool, in *Computer Vision–ECCV 2008*. An efficient dense and scale-invariant spatio-temporal interest point detector (Springer, 2008), pp. 650–663
19. V Kellokumpu, G Zhao, M Pietikäinen, Recognition of human actions using texture descriptors. Mach. Vis. Appl. **22**(5), 767–780 (2011)
20. S-R Ke, HLU Thuc, Y-J Lee, J-N Hwang, J-H Yoo, K-H Choi, A review on video-based human activity recognition. Computers. **2**(2), 88–131 (2013)
21. JK Aggarwal, MS Ryoo, Human activity analysis: a review. ACM Comput. Surv. (CSUR). **43**(3), 16 (2011)
22. R Poppe, A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
23. H Xu, Q Tian, Z Wang, J Wu, A survey on aggregating methods for action recognition with dense trajectories. Multimedia Tools Appl. **75**(10), 1–17 (2015)
24. DD Dawn, SH Shaikh, A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector. Vis. Comput. **32**(3), 1–18 (2015)
25. I Laptev, T Lindeberg, in *IN ICCV*. Space-time interest points (Springer, 2003), pp. 432–439
26. C Schüldt, I Laptev, B Caputo, in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference On*. Recognizing human actions: a local SVM approach, vol. 3 (IEEE, 2004), pp. 32–36
27. I Laptev, M Marszałek, C Schmid, B Rozenfeld, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. Learning realistic human actions from movies (IEEE, 2008), pp. 1–8
28. A Klaser, M Marszałek, C Schmid, in *BMVC 2008-19th British Machine Vision Conference*. A spatio-temporal descriptor based on 3d-gradients (British Machine Vision Association, 2008), pp. 275–1
29. H Kuehne, H Jhuang, E Garrote, T Poggio, T Serre, in *Computer Vision (ICCV), 2011 IEEE International Conference On*. Hmdb: a large video database for human motion recognition (IEEE, 2011), pp. 2556–2563

30. F Baumann, A Ehlers, B Rosenhahn, J Liao, Recognizing human actions using novel space-time volume binary patterns. Neurocomputing. **173**, 54–63 (2016)
31. MAR Ahad, T Ogata, J Tan, H Kim, S Ishikawa, A complex motion recognition technique employing directional motion templates. Int. J. Innov. Comput. Inf. Control. **4**(8), 1943–1954 (2008)
32. L Yeffet, L Wolf, in *Computer Vision, 2009 IEEE 12th International Conference On*. Local trinary patterns for human action recognition (IEEE, 2009), pp. 492–497
33. C-C Chen, J Aggarwal, in *Motion and Video Computing, 2009. WMVC'09. Workshop On*. Recognizing human action from a far field of view (IEEE, 2009), pp. 1–7
34. Z Gao, G Lu, P Yan, in *Digital Signal Processing (DSP), 2016 IEEE International Conference On*. Enhancing action recognition in low-resolution videos using dempster-shafer's model (IEEE, 2016), pp. 676–680
35. S Rahman, J See, in *Computational Science and Engineering (ICCSE), 2016 International Conference On*. Deep CNN object features for improved action recognition in low quality videos, (2016)
36. MAR Ahad, J Tan, H Kim, S Ishikawa, A simple approach for low-resolution activity recognition. Int. J. Comput. Vis. Biomech. **3**(1), 17–24 (2010)
37. X Wang, L Wang, Y Qiao, in *Computer Vision–ACCV 2012*. A comparative study of encoding, pooling and normalization methods for action recognition (Springer, 2013), pp. 572–585
38. C Harris, M Stephens, in *Alvey Vision Conference*. A combined corner and edge detector, vol. 15, (1988), p. 50
39. G Farnebäck, in *Image Analysis*. Two-frame motion estimation based on polynomial expansion (Springer, 2003), pp. 363–370
40. H Bay, T Tuytelaars, L Van Gool, in *Computer vision–ECCV 2006*. SURF: speeded up robust features (Springer, 2006), pp. 404–417
41. MA Fischler, RC Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM. **24**(6), 381–395 (1981)
42. H Wang, A Kläser, C Schmid, C-L Liu, Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis. **103**(1), 60–79 (2013)
43. T Ahonen, A Hadid, M Pietikainen, Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
44. V Ojansivu, Heikkilä, in *Image and Signal Processing*. Blur insensitive texture classification using local phase quantization (Springer, 2008), pp. 236–243
45. J Kannala, E Rahtu, in *Pattern Recognition (ICPR), 2012 21st International Conference On*. Bsif: Binarized statistical image features (IEEE, 2012), pp. 1363–1366
46. A Hyvärinen, J Karhunen, E Oja, *Independent Component Analysis, vol. 46*. (Wiley, 2004)
47. J Päivärinta, E Rahtu, J Heikkilä, in *Image Analysis*. Volume local phase quantization for blur-insensitive dynamic texture classification (Springer, 2011), pp. 360–369
48. J Liu, J Luo, M Shah, in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference On*. Recognizing realistic actions from videos "in the wild" (IEEE, 2009), pp. 1996–2003
49. T Wiegand, GJ Sullivan, G Bjøntegaard, A Luthra, Overview of the h. 264/avc video coding standard. IEEE Trans. Circ. Syst. Video Technol. **13**(7), 560–576 (2003)
50. A Vedaldi, A Zisserman, Efficient additive kernels via explicit feature maps. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 480–492 (2012)
51. K Simonyan, A Zisserman, Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
52. C Yan, Y Zhang, J Xu, F Dai, J Zhang, Q Dai, F Wu, Efficient parallel framework for hevc motion estimation on many-core processors. IEEE Trans. Circ. Syst. Video Technol. **24**(12), 2077–2089 (2014)