On the Effects of Low Video Quality in Human Action Recognition

John See
Faculty of Computing and Informatics
Multimedia University
Cyberjaya, Selangor, Malaysia
Email: johnsee@mmu.edu.my

Saimunur Rahman
Faculty of Computing and Informatics
Multimedia University
Cyberjaya, Selangor, Malaysia
Email: saimunur.rahman14@student.mmu.edu.my

***Abstract*—Human activity recognition is one of the most intensively studied areas of computer vision and pattern recognition in recent years. A wide variety of approaches have shown to work well against challenging image variations such as appearance, pose and illumination. However, the problem of low video quality remains an unexplored and challenging issue in real-world applications. In this paper, we investigate the effects of low video quality in human action recognition from two perspectives: videos that are poorly sampled spatially (low resolution) and temporally (low frame rate), and compressed videos affected by motion blurring and artifacts. In order to increase the robustness of feature representation under these conditions, we propose the usage of textural features to complement the popular shape and motion features. Extensive experiments were carried out on two well-known benchmark datasets of contrasting nature: the classic KTH dataset and the large-scale HMDB51 dataset. Results obtained with two popular representation schemes (Bag-of-Words, Fisher Vectors) further validate the effectiveness of the proposed approach.**

I. Introduction

Human action recognition in video is an active area of research, with many real-world applications ranging from automated video surveillance, video mining and retrieval, and interactive video games. In unconstrained videos, human actions are captured with typical image variations such as appearance, scale and view pose, to more challenging problems such as illumination change, occlusion, shadow, and camera motion. One relatively unexplored problem is pertaining the quality of videos. As most research in action recognition rely on the assumption that video data is of high-definition (HD) quality with minimal signal noise, many proposed approaches have found to worked well under such pristine conditions. However, most of these videos are not feasible for real-time video processing, data streaming and mobile applications, due to the computational overhead that most methods incur.

Popular approaches for generic image classification have been extended for use in video sequences, to a good measure of success. In particular, bag-of-words (or bag-of-features) based methods have demonstrated promising results for the task of action recognition [1]–[3]. Even with these successes, the representation of local regions in videos is still an open problem in research. As such, a variety of spatio-temporal features have also been considered in literature to better represent video data. Many popular works [1], [4], [5] prefer utilizing gradient and flow information to describe the shape and motion that lies in the video. The use of textures, however, is less common [6], [7], though there are many benefits that can be leveraged.

Oh et al. [8], in establishing the recent large-scale VIRAT dataset for continuous surveillance, provided nine downsampled versions of the data in the initial version[1]), consisting of three spatial scales and three temporal frame rates. The authors stressed that this is a ”relatively unexplored area” and that ”it is important to understand how existing approaches will behave differently”. Hence, this motivates this work to investigate the capability of recognizing actions under these challenging conditions.

Inspired by the known merits of various features and the obvious lack of action recognition work in low quality videos, we intend to investigate and present a feasible approach to this problem. In this paper, we propose the usage of spatio-temporal textural features as a complement to shape and motion features in order to increase the robustness of recognizing human actions under such conditions. We approach this work by examining two forms of low quality video: videos that are poorly sampled spatially (low resolution) and temporally (low frame rate), and compressed videos that are affected by motion blurring and compression artifacts. We evaluate the proposed approach with a set of extensive experiments on two well-known benchmark action datasets of contrasting nature: the KTH dataset, which contains simple actions under controlled environment; and the large-scale HMDB51 dataset, which consists of video clips captured from movies and YouTube under complex, unconstrained environments. Finally, we also provide an analysis into the aspects of descriptor sampling size, encoding methods, and computational cost.

The rest of the paper is organized as follows: Section II briefly reviews some related work in literature; Section III defines how video downsampling is carried out in our work; Section IV describes the steps involved in the overall framework; Section V presents the datasets used, experimental results and further analysis on various aspects; Finally, Section VI concludes the paper and suggests some future directions.

[1] As of today, these downsampled versions are no longer available in the current VIRAT version 2.0. Website: http://www.viratdata.org/

II. Related Work

In the last decade, human action recognition has been studied extensively by the computer vision and pattern recognition community [9], [10]. From the the recent research in activity recognition, spatio-temporal video features can be categorized into three main categories based on the type of feature used: dynamic feature (motion), structure (shape) and textural (texture), or any implicit/explicit combination of these three types. Most recent works employ primarily motion and shape features [3]. Laptev [11] first proposed the extraction of shape (HOG) and motion (HOF) information from spatio-temporal interest points (STIP) to classify human actions in video. More recently, Wang et al. [5] proposed the use of dense trajectories with the same way of encoding the shape and motion information. All these methods appear to suggest that the combination of shape and motion features performs better than using them individually. Some recent works focused on other aspects of action recognition besides feature extraction, such as the use of human body parts [12], [13], encoding techniques [14], [15] and deep learning methods [16], [17].

Texture-based (or textural) features have also found their way to action recognition research, particularly the spatio-temporal descriptor LBP-TOP [18], a spatio-temporal extension to the well-known Local Binary Pattern (LBP) descriptor first proposed by Ojala et al. [19]. Kellokumpu et al. [6] first proposed the use of the LBP-TOP descriptor to recognize human actions by applying it on the entire bounding volume area. A histogram of local textural features is then built from the spatio-temporal data from the entire action sequence. Their experiments on the Weizmann dataset showed tremendous promise in comparison with popular approaches back then.

Mattivi and Shao [7] applied LBP-TOP over small video patches called cuboids which are extracted from each interest point detected from the sequence. This produces a sparser representation of video sequences unlike approaches that utilize the whole video volume. The LBP-TOP cuboids are then passed through a typical bag-of-words classification scheme. Their approach, after a few additional improvements, managed a promising accuracy rate of around 91% on the KTH dataset. A subsequent work by the same authors [20] further verified the strength of LBP-TOP features. The main disadvantage of this approach is that the textural features are too dependent on the extraction of spatio-temporal interest points (in this case, the cuboids). It remains to be seen how cuboids would fare in circumstances of video quality.

Yeffet and Wolf [21] proposed a self-similarity approach within an efficient representation, which is motivated by LBP. Their concept simply compare a small patch of pixels with shifted patches in the previous and in the next frame. Hence, the encoded information, named Local Trinary Pattern (LTP), describes the relative similarity of the said two patches to the patch in the central frame. Histograms are accumulated every few frames and the vector of all concatenated histograms is taken as the video descriptor. Their extensive experiments on both simple and complex action datasets showed excellent results, though the authors acknowledged the lack of appearance information in the absence of motion.

III. Video Downsampling

A video's spatial resolution and temporal sampling rate defines the amount of spatial and temporal information it can convey. Spatial resolution is simply the video's horizontal pixel count by its vertical pixel count, i.e. frame size. The temporal sampling rate defines the number of discrete frames in a unit of time, i.e. frames per second (fps) or Hertz (Hz). Based on our recent work [22], we first describe how spatial and temporal downsampling is employed to create several downsampled versions of the original video data

A. Spatial Downsampling

Spatial downsampling produces an output video with a smaller resolution than the original video. In the process, no additional data compression is applied while the frame rates remained the same. For clarity, we define a spatial downsampling factor, α which indicates the factor in which the original spatial resolution is reduced. In this work, we fixed $\alpha = \{2, 3, 4\}$ for modes SD_α, denoting that the original videos are to be downsampled to half, a third and a fourth of its original resolution respectively. Fig. 1 shows a sample video frame that undergoes SD_2, SD_3 and SD_4. We opted not to go beyond $\alpha = 4$ as extracted features are too few and sparse to provide any meaningful representation.

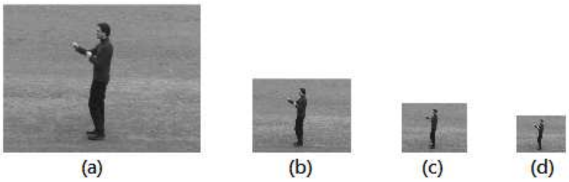

Fig. 1. Spatially downsampled videos. (a) Original (SD_1); (b) SD_2; (c) SD_3; (d) SD_4;

B. Temporal Downsampling

Temporal downsampling produces an output video with smaller temporal sampling rate (or frame rate) than the original video. In the process, the video frame resolution remained the same. Likewise, we also define a temporal downsampling factor, β which indicates the factor in which the original frame rate is reduced. Fig. 2 illustrates how the frames are selected in the downsampling process; black strips indicate the kept frames while orange strips indicate the discarded frames. In our work, we use values of $\beta = \{2, 3, 4\}$ for modes TD_β, denoting that the original videos are to be downsampled to half, a third and a fourth of its original frame rate respectively.

IV. Overall Framework

Figure 3 shows the overall framework of the proposed approach for action recognition. We now describe the steps involved in the framework in detail.

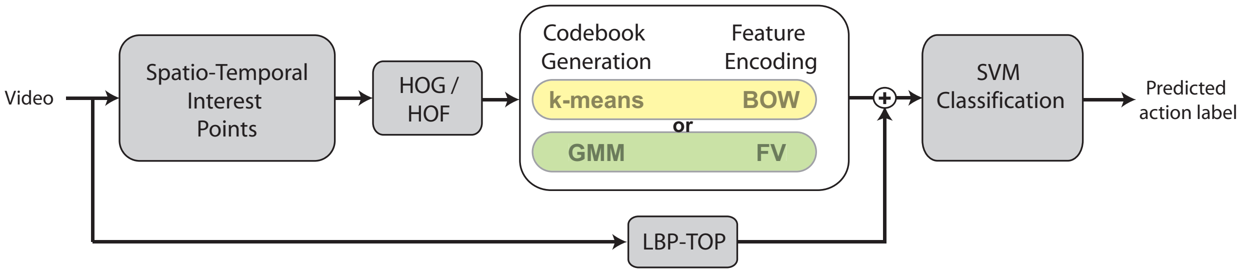

Fig. 3. Overall framework of the proposed action recognition scheme

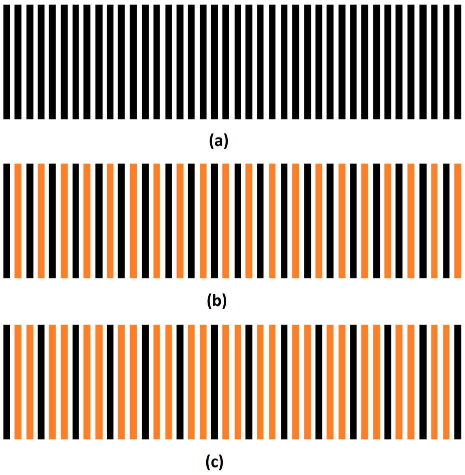

Fig. 2. Temporal downsampling; (a) Original video (b) TD_2; (c) TD_3;

A. Spatio-temporal Interest Points

For each given sample point (x, y, t, σ, τ), a feature descriptor is computed for a 3-D video patch centered at (x, y, t) at spatial scale σ and temporal scale τ; hence these points can be defined in five dimensions (x, y, t, σ, τ).

In this work, we employ the Harris3D detector (a space-time extension of the popular Harris detector [23]) to obtain spatio-temporal interest points (STIP) [11]. Briefly, a spatio-temporal second-moment matrix is computed at each video point $\mu(.; \sigma; \tau) = g(.; s\sigma; s\tau) * (\nabla L(.; \sigma; \tau)L(.; \sigma; \tau))^T$ using a separable Gaussian smoothing function g, and space time gradients ∇L. The final location of the detected STIPs are given by local maxima of $H = \det(\mu) - k\ \text{trace}^3(\mu), H > 0$. We use the implementation from [1] with standard parameter settings [3], i.e. $k = 0.00005$, $\sigma^2 = \{4, 8, 16, 32, 64, 128\}$ and $\tau^2 = \{2, 4\}$, for both original and downsampled videos. For a small portion (5%) of video clips that undergo spatial and temporal downsampling to much greater extent $(\alpha = 4, \beta = 3, 4)$, increasing the value of k can impose more sensitivity towards the change in gradients, thus picking up more STIPs. Figure 4 shows some STIPs that were extracted using the Harris3D detector from two types of actions.

B. Local Descriptors

Following the extraction of STIPs, local descriptors are computed at these STIP locations to capture spatio-temporal features that characterizes the action in the clip. Next, we will briefly discuss the shape, motion and texture features used in our work.

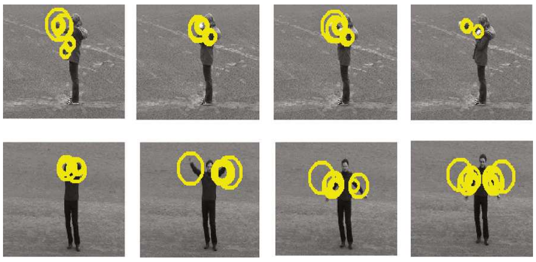

Fig. 4. Harris3D feature detector on KTH data set

1) Shape and Motion Features:

In general, *shape* features depict the structural or geometrical information found spatially in video; *motion* features carry important temporal information or changes of its structure across time. These two features can be taken together to exemplify spatio-temporal information in video. An extensive experimental work by Wang et al. [3] highlighted the strengths of *shape* and *motion* features when used together as spatio-temporal features.

To characterize the local shape and motion information accumulated in space-time neighborhoods of the detected STIPs, we extract Histogram of Gradient (HOG) and Histogram of Optical Flow (HOF) descriptors as proposed in [3], [11]. The HOG/HOF descriptors are computed for the interest points by defining descriptor volumes of size $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma, \Delta_t(\tau) = 8\tau$. Each volume is subdivided into a $n_x \times n_y \times n_t$ grid of cells; for each cell, 4-bin histograms of gradient orientations (HOG) and 5-bin histograms of optical flow (HOF) are computed. In this experiment we opted for grid parameters $n_x, n_y = 3, n_t = 2$ for all videos, as suggested in Wang et al. [3].

2) Texture Features:

Textures are defined as statistical regularities that describe patterns found in both space and time, which was adopted for action recognition with promising results [6], [7].

One of the most widely-used texture descriptor, Local Binary Pattern (LBP) produces a binary code at each pixel location by thresholding pixels within a circular neighborhood region by its center pixel [19]. The $LBP_{P,R}$ operator produces

2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighborhood set. After computing these LBP patterns for the whole image, an occurrence histogram is constructed to provide a statistical description of the distribution of local textural patterns in the image. This descriptor has been proved to be successful in face recognition [24]. In order to be applicable to the temporal domain where textures are viewed as "dynamic patterns" that change over time, Zhao et al. [18] proposed LBP on Three Orthogonal Planes (LBP-TOP), where LBP is performed on the three orthogonal planes (XY, XT, YT) in the video volume by concatenating their respective occurrence histograms into a single histogram. LBP-TOP is formally expressed by $LBP-TOP_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_Z}$ where the subscripts denote a neighborhood of P points equally sampled on a circle of radius R on XY, XT and YT planes respectively. The resulting feature vector length is (3×2^P). LBP-TOP encodes the appearance and motion along three directions, incorporating spatial information in XY-LBP and spatial temporal co-occurrence statistics in XT-LBP and YT-LBP. In this experiment we apply the parameter settings of $LBP-TOP_{8,8,8,2,2,2}$ with non-uniform patterns as specified by Mattivi and Shao [7], which produces a feature vector length of 768.

C. Codebook Generation

A video sequence is represented as a bag of local spatio-temporal features [25]. Spatio-temporal features are first quantized into a number of visual words called a *codebook*, and then the video is represented by the frequency histogram of these visual words.

In this work, we tested with two representation schemes – Bag-Of-Words (BOW) and Fisher Vectors (FV), each of which generates its own codebook to encode the feature descriptors. Codebooks can be constructed from a single, or a combination of different feature descriptors concatenated at the descriptor level. Alternatively, separate codebooks can also be constructed for each feature descriptor before concatenation, at the expense of more computational load. For both representations, we sample a subset of 100,000 randomly selected descriptors from the training samples to limit the complexity of our experiments. This is a typical setting found to give a reasonably good and stable performance across datasets [3].

1) Bag-of-Words (BOW): Codebooks are generated with standard k-means clustering. For consistency, we empirically set the number of visual words to $V = 4000$, which has shown to give good results in numerous works [3]. Naturally, feature encoding is performed by vector quantization (VQ), which is a hard assignment scheme based on the nearest Euclidean distance. A feature is assigned to cluster c if it's closer to the centroid of cluster c than any other centroids. To increase the clustering precision, we iteratively perform k-means 8 times (initialized using the preceding round's final centers) and kept the result with the lowest error.

2) Fisher Vectors (FV): We construct Fisher Vectors for each type of descriptor separately. In this representation technique, a Gaussian Mixture Models (GMM) is first fitted to the set of descriptors that were randomly selected from the training samples. Given the GMM fitting $\theta = (\pi_j, \mu_j, \Sigma_j)$ where the parameters π_j, μ_j and Σ_j indicate the prior probability, mean and covariance of each j-th distribution, the GMM associates each descriptor x_i to a mode j in the mixture.

Fisher Vectors can be encoded by the mean (μ_{jk}) and standard deviation (σ_{jk}) vectors for each mode k,

$$\mu_{jk} = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} p_{ik} \frac{x_{ji} - \mu_{ik}}{\sigma_i} \tag{1}$$

$$\sigma_{jk} = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} p_{ik} [(\frac{x_{ji} - \mu_{ik}}{\sigma_i})^2 - 1] \tag{2}$$

where $j \in \mathbb{R}^D$ and the posterior probability defined by

$$p_{ik} = \frac{(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}{\sum_{t=1}^{K} (x_i - \mu_t)^T \Sigma_t^{-1} (x_i - \mu_t)} \tag{3}$$

constitutes the soft assignment of local descriptor x_i to the k-th Gaussian mixture mode. Unlike hard assignment, this technique gives a probability measure to each assigned descriptor, providing also the shape of the distribution.

The Fisher Vectors η_{jk} are obtained by concatenating the vectors μ_{jk} and σ_{jk} for all K modes in the Gaussian mixtures, yielding a final encoding dimension of $2KD$. Finally, the FVs are then power-normalized by the following function

$$f(\eta_j) = sign(\eta_j)|\eta_j|^\alpha \tag{4}$$

with $\alpha = 0.5$ then ℓ_2-normalized, following [26]. In our experiments, we use $K = 256$ for each descriptor type.

D. Classification

For classification, we use a non-linear support vector machine (SVM) with a χ^2-kernel which was used in [1]:

$$K(H_i, Hj) = \exp(-\frac{1}{2A} \sum_{n=1}^{K} \frac{(h_{in} - h_{jn})^2}{h_{in} - h_{jn}}) \tag{5}$$

where h_{in} and h_{jn} are the frequency histograms of the n-th word occurrences, K is the vocabulary size, and A is the mean value of distances between all training samples. This is neatly approximated by Vedaldi et al. [27] in the form of additive homogeneous kernels which are more efficient and accurate. In some parts of our experiments where FV is used, we opted for a linear kernel instead of χ^2 kernel, which is known to over-fit feature vectors of higher dimensionality. For multi-class SVM classification, we apply the *one-versus-all* (OVA) approach to select the class with the highest score.

V. Experiments

In this section, we describe a set of extensive experiments and their respective results, while analyzing and comparing different combination of feature descriptors discussed earlier. Experiments were conducted separately for spatial downsampling and temporal downsampling to demonstrate the strengths of specific features with respect to each condition. We also provide a detailed elaboration of the evaluation framework and settings used for each experimented dataset.

A. Datasets

We conducted our experiments on two notable benchmark datasets: KTH [25] and HMDB51 [28]; the former being a classic dataset that is most popular in action recognition research, the latter being a large-scale dataset with videos captured "in the wild". Both datasets are very contrasting in terms of the environment in which the videos were captured in, the extent of camera motion and view changes, and the number of action classes. The HMDB51 is of great appeal to our work as it provides specific quality labels for all videos. As such, we did not consider other contemporary large scale action datasets (UCF50, UCF101) as they neither specified video quality labels, nor low quality subsets. An extensive amount of downsampling work and feature extraction is required to test with these datasets.

KTH [25] is the most popular dataset in literature for human action recognition. It contains 6 action classes: walking, running, jogging, hand-waving, hand-clapping and boxing; performed by 25 actors in 4 different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. There are 599 video samples in total (one subject has less one clip). Each clip is sampled at 25 fps and lasts between 10–15 seconds with image frame resolution of 160 × 120 pixels. We follow the original experimental setup, i.e., the samples are divided into a test set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and training set (containing the remaining 16 subjects), while reporting the average accuracy over all classes as performance measure. For the purpose of this work, the six downsampled versions of the dataset (denoted by SD_2, SD_3, SD_4, TD_2, TD_3, TD_4; detailed explanation in Section III) are used. Figure 5 shows a sample frame from various spatially and temporaly downsampled video clips from the KTH dataset.

The **HMDB51** [28] dataset is a large scale action video database with 51 action categories[2] totaling 6,766 clips, extracted from a variety of sources ranging from digitized movies to YouTube. Video clips depict mainly natural actions from uncontrolled environments (i.e. "in the wild"), with a wide range of camera viewpoints, the presence of camera motion, a highly variable number of humans involved in the action. Each category contains at least 101 clips. In addition to the action labels, each clip is also annotated with meta-labels describing various properties of the clip including the quality of video. A three level grading of video quality was applied to the set of clips. A requirement was set to gauge the ease of observers in identifying single fingers during the motion. Video samples that do not meet this requirement were rated "medium" or "bad" if body parts or limbs vanish while the action is executed. In addition, the "bad" videos also contain significant motion blurring and compression artifacts. Figure 6 shows a sample frame from "bad" and "medium" quality video clips of various action classes from the HMDB51 dataset.

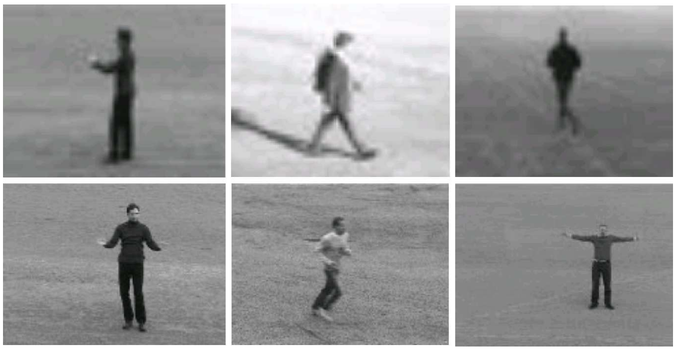

Fig. 5. Sample action classes from KTH: *Top row* (Spatially downsampled, shown here after resizing): Boxing, Walking, Jogging; *Bottom row* (Temporally downsampled): Handclapping, Running, Handwaving.

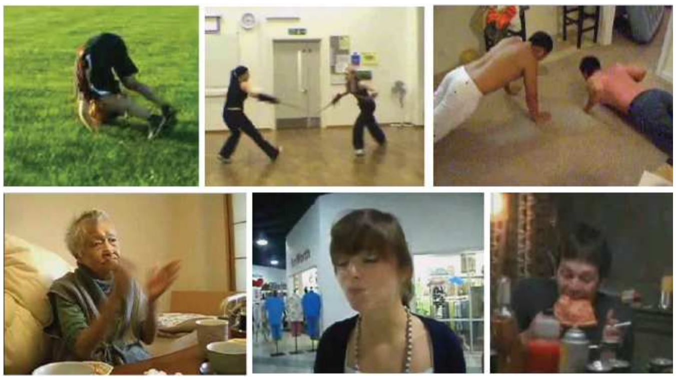

Fig. 6. Sample action classes from HMDB51: *Top row* (full body motion): Somersault, Fencing, Push-ups; *Bottom row* (motion from specific body parts or face): Clap, Chew, Eat. Frames shown here are from "bad" and "medium" labeled clips.

For the purpose of this work, we are mainly interested in the evaluation of clips annotated with "medium" and "bad" quality labels. However, the "high" quality clips are also useful as a control experiment for the sake of comparison. Hence, we partition the HMDB dataset into three subsets based on its quality label (distribution in parenthesis): **HMDB-BQ** (20.8%) containing "bad" quality clips, **HMDB-MQ** (62.1%) containing "medium" quality clips, and HMDB51-HQ (17.1%) comprising of the remaining "high" quality clips. For consistency of experiments, we follow the settings used in the original paper [28] whereby three distinct training-test splits (70/30 clip distribution per class) were used. The training sets remain the same, with a fair composition of videos from all three quality levels. The mean accuracy of all three splits are

[2] Data available at http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/.

TABLE I
RECOGNITION ACCURACY (%) OF VARIOUS FEATURE COMBINATIONS WITH BAG-OF-WORDS (BOW) AND FISHER VECTOR (FV) REPRESENTATION ON VARIOUS DOWNSAMPLED VERSIONS OF THE KTH DATASET.

Method	BOW (V=4000)						FV (K=256)					
	SD_2	SD_3	SD_4	TD_2	TD_3	TD_4	SD_2	SD_3	SD_4	TD_2	TD_3	TD_4
HOG	76.85	66.20	55.56	80.09	76.85	75.46	75.00	69.44	55.09	86.57	81.94	84.26
HOG+LBP-TOP	80.56	73.61	76.39	80.56	75.46	74.54	79.63	76.85	75.93	85.19	83.80	79.17
HOF	88.89	82.41	76.39	83.80	75.46	72.22	87.50	82.87	76.38	85.19	81.94	76.85
HOF+LBP-TOP	**89.35**	**85.65**	**84.26**	83.80	80.56	78.70	**88.43**	**82.87**	**81.94**	86.11	83.80	78.70
HOGHOF	83.33	76.39	65.74	86.11	81.94	76.85	86.11	80.09	64.35	88.43	84.26	82.87
HOGHOF+LBP-TOP	86.11	77.31	77.31	**89.35**	**85.65**	**81.94**	87.04	82.41	78.70	**90.28**	**85.19**	**84.72**

reported as the final measure of performance.

B. Experimental Results

We first present results from two comprehensive experiments – one on the spatially and temporally downsampled clips (in controlled environment, based on the KTH), and the second on low quality clips compromised by motion blurring and compression artifacts (in uncontrolled environment, from the HMDB). Further to that, we also provide a detailed analysis into various factors relating to the performance of our proposed approaches.

TABLE II
RECOGNITION ACCURACY (%) OF DIFFERENT FEATURE COMBINATIONS WITH BAG-OF-WORDS (BOW) AND FISHER VECTOR (FV) REPRESENTATION ON HMDB-MQ AND HMDB-BQ SUBSETS.

Method	HMDB-BQ		HMDB-MQ	
	BoW	FV	BoW	FV
HOG+HOF	16.44	21.57	22.87	30.79
HOGHOF+LBP-TOP	23.48	**28.66**	28.32	33.94
HOG+HOF+LBP-TOP	**26.04**	28.49	**30.99**	**35.24**
HOGHOF (Baseline) [28]	17.18	-	18.68	-
C2 (Baseline) [28]	17.54	-	23.10	-
LBP-TOP	17.00		24.11	

1) Experiment I – Downsampled videos: Due to the tedious nature of this experiment (which requires creating six downsampled versions of a single dataset), we ran our experiments only on the classic KTH dataset. Each type of descriptor (**HOG**, **HOF** and **HOGHOF**), and its concatenation with the spatio-temporal textural feature of LBP-TOP (**HOG+LBP-TOP**, **HOF+LBP-TOP**, **HOGHOF+LBP-TOP**) were evaluated across all six downsampled versions (SD_2, SD_3, SD_4, TD_2, TD_3, TD_4) on two encoding techniques (BOW and FV). HOGHOF denotes concatenation of HOG and HOF descriptors before codebook generation, as seen in [3]; HOG+HOF denotes concatenation of the codebooks of both HOG and HOF features.

Results in Table I show some interesting observations. As the video clip deteriorates spatially, the motion features (HOF) appear to be most robust and able to sustain a sufficiently high accuracy with the help of textural features (LBP-TOP). For instance, there is only a $\sim 5\%$ drop from SD_2 to SD_4 for BOW when LBP-TOP is used, in contrast to $> 12\%$ drop without LBP-TOP. The relevancy of motion information makes sense since the shape information (gradient) is largely dependent on the change of intensities in the spatial domain. On the other hand, both shape and motion features (HOGHOF) are necessary to maintain a reasonably good accuracy in the face of temporal downsampling, even more so when textural features (LBP-TOP) are considered together as well. For instance, the drop in accuracy from TD_2 to TD_4 is limited to just $< 6\%$ for FV encoding when LBP-TOP is used. Overall, these characteristics are apparent in both BOW and FV encoding methods. Interestingly, the FV encoding generally performs better for the temporally downsampled videos compared to the spatially downsampled videos.

Shape feature (HOG) performed generally poorer than the other feature combinations. We observed that shape information becomes less discriminant as spatial resolution decreases. However, in the case of decreasing temporal frame rate, it can sustain a reasonable recognition rate.

2) Experiment II – Low quality videos from HMDB51: In the second experiment, we evaluated the methods on the HMDB-BQ (bad quality) and HMDB-MQ (medium quality) subsets, on both encoding techniques (BOW and FV). The codebook size for BoW is set to $V = 8000$ while $K = 256$ for FV. Table II shows the performance of the evaluated methods, in contrast to the baseline performances reported in the dataset paper [28]. Clearly, the STIP-based descriptors that were aided by the robustness of the LBP-TOP descriptor seemed to produce a significant leap of improvement in performance over their normal counterparts. Figure 8 illustrates this in better detail, highlighting the breakdown of improvement by the three subsets ('bad','medium', 'good'). Recognition of actions in the bad quality videos (HMDB-BQ subset) was vastly better, with a top improvement of around 70% in the case of HOGHOF under BoW encoding. It is interesting to mention that the LBP-TOP descriptor itself performed surprisingly better than the two baseline methods in the medium quality subset.

Figure 7 compares the confusion matrices of the HOG+HOF and HOG+HOF+LBP-TOP methods (without and with textural information, respectively), obtained from the first split, with BoW encoding. The HOG+HOF+LBP-TOP combination of descriptors clearly show larger diagonal values in numerous classes. Altogether, there were 20 classes that had improved its recognition accuracy while only 9 classes had a drop in accuracy; the rest remained unchanged. Among those action classes that had improved (by more than 50%) are such as ”Pull-up”, ”Push-up” and ”Chew”.

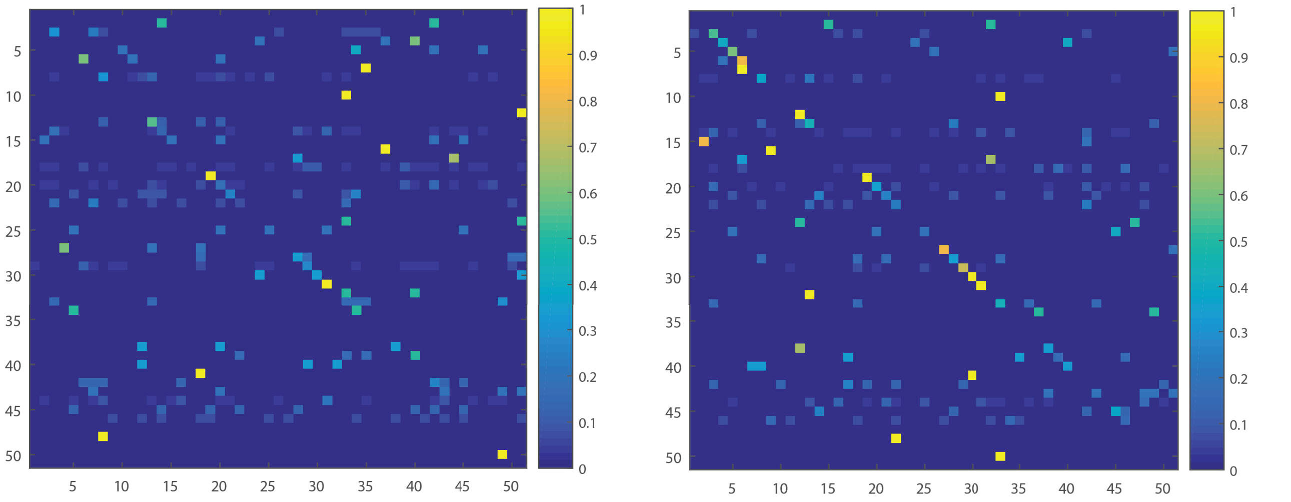

Fig. 7. Confusion matrices for HMDB51 using HOG+HOF *(left)* and HOG+HOF+LBP-TOP *(right)* with FV representation. Best viewed in colour.

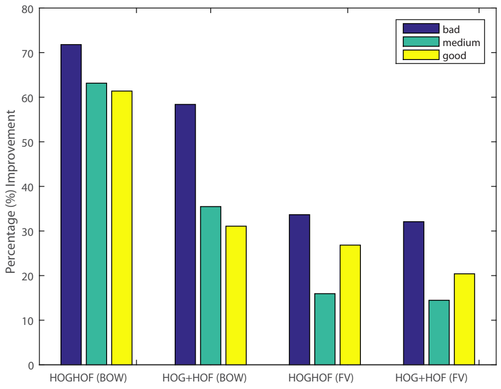

Fig. 8. Percentage (%) of increment after inclusion of textural features (LBP-TOP) for various configurations.

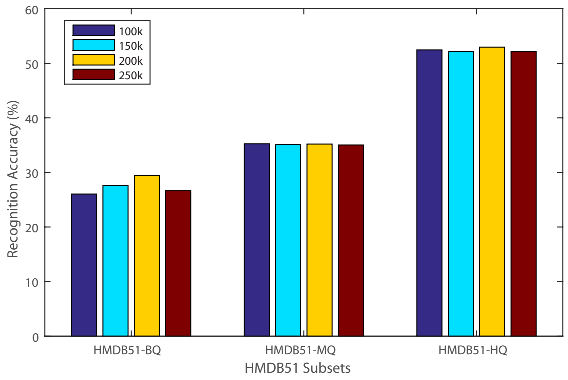

Fig. 9. Recognition accuracy (%) of HOG+HOF+LBP-TOP (FV) approach on the HMDB51 subsets with respect to the number of randomly sampled descriptors.

3) Analysis on Random Sampling Size: Many works in literature have analyzed the impact of selecting different codebook sizes [14], [15], and other factors such as the choice of normalization and pooling methods [14]. In our work, many of these parameters were chosen based on practical suggestions or specific values recommended by these authors. For instance, due to the complexity of movements in the HMDB51 clips, we have chosen $K = 8000$ instead of $K = 4000$ which suffice for the simpler KTH clips. Nevertheless, one aspect remains unexplored, that is the number of feature descriptors randomly selected to build the codebook. In the previous two experiments, we set this value to 100,000 for consistency in experiments. It can be expected that using larger number of samples will result in a more stable performance, but at the expense of heavier computational load. Figure 9 shows the performance of the best proposed approach (HOG+HOF+LBP-TOP) on the three HMDB51 subsets, with respect to the number of randomly sampled descriptors. It can be seen that the ’bad’ quality subset would fare much better using around 200,000 randomly sampled descriptors to build the codebook. This factor is not as significant in the other two subsets.

4) Analysis on Encoding Methods: The choice of encoding method as seen in the experimental results of various evaluation works [14], [29] seemed to show a distinct advantage of Fisher Vector (FV) over other encoding schemes such as sparse encoding and histogram encoding (used in BoW). It is interesting to note (from Table I) that the FV encoding does not hold any advantage over BoW encoding as the spatial resolution decreases (in fact, it loses in some cases). When temporal frame rate deteriorates, FV is clearly better than BoW. In the HMDB51 experiment (see Table II), the FV outperforms the BoW representation on all accounts.

5) Analysis on Computational Cost: Experiments were carried out on an Intel Core-i7 3.6 GHz machine with 24GB

RAM. The incorporation of LBP-TOP has a negligible effect on the time taken for codebook generation, the heaviest task in the recognition pipeline. With the homogeneous kernel map [27], feature dimension for LBP-TOP is $\ell_L = (2^P \cdot 3 \cdot 3)$ or three times larger (which works out to be 2304 since we use $P = 8$ without uniform patterns). This is still much smaller than the feature dimension of the STIP-based descriptors, i.e. $\ell_L \ll \ell_{STIP}$ which is V for BoW, or $2DK$ for FV.

However, LBP-TOP has a computational complexity of $\mathcal{O}(XYT \cdot 2^P)$, with X, Y denoting the frame resolution, and T is the number of frames; this expensive feature extraction process remains the most inherent drawback when the addition of LBP-TOP is considered. It is worth mentioning that the LBP-TOP feature extraction time also decreases exponentially with respect to the spatial downsampling factor (with X and Y only a fraction of the original).

VI. Conclusion

In this paper, we investigate the effects of low video quality in human action recognition. To the best of our knowledge, there are no existing systematic attempts to investigate the problem of video quality, which is highly relevant in many real-world applications. By our scope, we considered videos that are poorly sampled spatially and temporally, as well as videos adversely affected by motion blurring and compression artifacts. To alleviate the degradation of information in video data, we propose to complement the conventional shape and motion features with spatio-temporal textural features which describes the statistical distribution of patterns. This preliminary work draws some interesting observations as to how low quality videos can particularly benefit from textural information, considering that most new approaches tend to involve only shape and motion information as their choice of space-time features. In fact, analysis on the HMDB51 dataset showed that the "bad" quality clips responded strongest across all tested settings with the inclusion of textural features.

In future, we intend to explore other textural features that are potentially more robust towards deterioration of video quality while also less computational expensive. Textural features that are denser [30] or richer in description [21] are also potential directions following this work.

VII. Acknowledgment

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

References

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE CVPR*, 2008, pp. 1–8.

[2] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *IJCV*, vol. 79, no. 3, pp. 299–318, 2008.

[3] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009, pp. 124–1.

[4] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.

[5] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[6] V. Kellokumpu, G. Zhao, and M. Pietikäinen, "Human activity recognition using a dynamic texture based method." in *BMVC*, 2008.

[7] R. Mattivi and L. Shao, "Human action recognition using lbp-top as sparse spatio-temporal feature descriptor," in *Computer Analysis of Images and Patterns*. Springer, 2009, pp. 740–747.

[8] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *IEEE CVPR*. IEEE, 2011, pp. 3153–3160.

[9] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[10] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013.

[11] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[12] O. R. Murthy, I. Radwan, A. Dhall, and R. Goecke, "On the effect of human body parts in large scale human behaviour recognition," in *Int. Conf. on DICTA*, 2013, pp. 1–8.

[13] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *IEEE ICCV*, 2013, pp. 3192–3199.

[14] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," in *Computer Vision–ACCV 2012*. Springer, 2013, pp. 572–585.

[15] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1817–1824.

[16] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*. Springer, 2011, pp. 29–39.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE CVPR*, 2014, pp. 1725–1732.

[18] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. PAMI*, vol. 29, no. 6, pp. 915–928, 2007.

[19] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. PAMI*, vol. 24, no. 7, pp. 971–987, 2002.

[20] L. Shao and R. Mattivi, "Feature detector and descriptor evaluation in human action recognition," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 2010, pp. 477–484.

[21] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 492–497.

[22] S. Rahman, J. See, and C. C. Ho, "Action recognition in low quality videos by jointly using shape, motion and texture features," in *IEEE Int. Conf. on Signal and Image Processing Applications*, 2015, p. To appear.

[23] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of 4th Alvey Vision Conference*, vol. 15, 1988, p. 50.

[24] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. PAMI*, vol. 28, no. 12, pp. 2037–2041, 2006.

[25] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Int. Conf. on Pattern Recognition*, vol. 3, 2004, pp. 32–36.

[26] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 143–156.

[27] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 480–492, 2012.

[28] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *IEEE ICCV*, 2011, pp. 2556–2563.

[29] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods." in *BMVC*, vol. 2, no. 4, 2011, p. 8.

[30] J. Ylioinas, A. Hadid, Y. Guo, and M. Pietikäinen, "Efficient image appearance description using dense sampling based local binary patterns," in *Computer Vision–ACCV 2012*. Springer, 2013, pp. 375–388.