Multimed Tools Appl (2018) 77:26635–26655
https://doi.org/10.1007/s11042-018-5882-z

Deep Event Learning boosT-up Approach: DELTA

Krishan Kumar[1,2] · **Deepti D. Shrimankar**[1]

Received: 22 March 2017 / Revised: 19 February 2018 / Accepted: 13 March 2018 /
Published online: 28 March 2018

Abstract Nowadays, the video surveillance systems may be omnipresent, but essential for supervision everywhere, e.g., ATM, airport, railway station and other crowded situations. In the multi-view video systems, various cameras are producing a huge amount of video content around the clock which makes it difficult for fast browsing, retrieval, and analysis. Accessing and managing such huge data in real time becomes a real challenging task because of inter-view dependencies, illumination changes and the bearing of many inactive frames. The work highlights an accurate and efficient technique to detect and summarize the event in multi-view surveillance videos using boosting, a machine learning algorithm, as a solution to the above issues. Interview dependencies across multiple views of the video are captured via weak learning classifiers in boosting algorithm. The light changes and still frames are tackled with moving an object in the frame by Deep learning framework. It helps to reach the correct decision for the active frame and inactive frame, without any prior information about the number of issues in a video. Target, as well as subjective ratings, clearly indicate the potency of our proposed DELTA model, where it successfully reduces the video data, while keeping the important information as events.

Keywords AdaBoosting · Deep learning · Event summarization · Key-frame · Multi-view video

✉ Krishan Kumar
kkberwal@nituk.ac.in

Deepti D. Shrimankar
dshrimankar@cse.vnit.ac.in

[1] Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology Nagpur, Nagpur, India

[2] National Institute of Technology Uttarakhand, Srinagar, India

1 Introduction

In this multimedia era, video surveillance is greatly affected due to the rapid increase in computing and networking base. It avails us to trim down the crimes by way of observing the victim, optimize the traffic mobility, and increase the transportation safety through the frequent use of the digital video technologies. Due to the utilization of these technologies, a huge amount of video data in the multi-view environment is generated by various applications such as sports, video surveillance and security systems, [27, 39, 40] etc. However, traditional video data access models are extensively limited for these audiovisual applications including, quick event recognition, suspicious activities and traffic patterns scrutiny, etc. Therefore, creating an efficient and compact comprehensive summary of long surveillance videos is a critical issue. Hence, the authors believe that a model is immediately required, which can focus on multi-view videos to summarize the events by capturing the significant activities in minimum time. Our target is to find the keyframes from the video by removal of unimportant frames where raw multi-view videos are the input, and a single summarization video will be generated as an end product.

Many supervised learning [11, 45, 54] as well as unsupervised Video Summarization (*VS*) models [6, 17, 30] are used for the event detection. Although, these methods seem time-consuming for the real-time applications. For mono-view videos, a large number of summarization models exist in the literature [4, 13, 23, 25, 32, 34, 36, 41]. Nevertheless, there are certain distinct challenges in the multi-view environment for VS than the mono-view. Multiple cameras are used to study the multi-view videos, where the entire recorded content may not be informative. Such complex content makes the VS process difficult to distill the useful information [26]. Moreover, the individual views are subjected to endure from the frequent light changes. In addition to this, the same event may be captured by multiple or all available cameras with a different point of the views. Thus, a big amount of interviews, statistical dependencies may exist in the multi-view videos [7]. So, mono-view VS models are not sound to handle the multi-view summarization problems as mentioned in [2, 7, 8, 22, 35, 42].

To address the above issues, this work highlights machine learning based AdaBoosting approach for the multiview environment, where correlations among different views are established through a combined Boosting classifier for resolving the intra-view redundancy. The optimized visual features are extracted from a video using AlexNet [20] as discussed in Fig. 1 for identifying the object status in a frame. The object status helps us to handle the illumination changes by removing the inactive frame or nonobject/ motion frame to achieve the accurate summary. Each ensemble takes care of the individual view of a video. Due to the execution of the individual view in parallel, the model can meet the requirement of real-time applications. Moreover, the DELTA model can detect the events from two or more

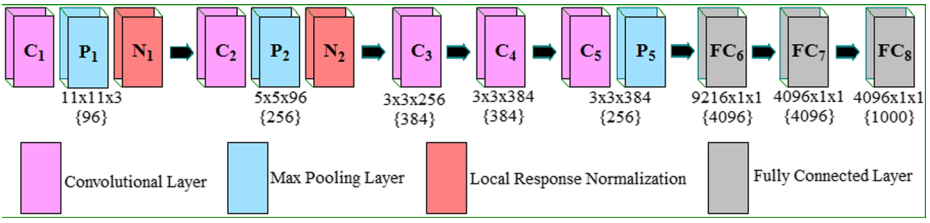

Fig. 1 An example of the AlexNet model developed by Krizhevsky et al. [20]

views in a surveillance system at a time, with the consideration of the correlation among different views. The proposed DELTA model is not entirely grasped the interesting events, but also locating the exact event boundary in real time. The remarkable contributions of the study are stated as follows:

- The multi-view VS problem is worked out by a machine learning based AdaBoosting approach. The AdaBoosting approach captures the inter-view dependencies with majority voting policy among n-views through n-trained classifiers more accurately and efficiently than the existing models.
- The trained classifiers are executed in parallel to preserve the computing time while the classifiers are trained only with the initial 10% frames not all. A deep learning framework is used to pull out the optimized features for identification of the moving objects in the frames, which aid us to handle the illumination changes by taking away moving object absence frame to reach an accurate summary. The Alex Net helps us to detect the moving object in the frame using cosine similarity between two consecutive frames, which is regarded in the processing of frame through the trained classifier. The combined classifier makes fast execution by compiling only motion/ active frames. Therefore, the processing time about 4.3 milliseconds per frame of the model also indicates that it meets the requirement of real-time applications.
- A major concern in Event Summarization is to find the truthful integrated model, including a location of the detected event boundaries which is achieved through Delta Model (Algorithm 2). The model also states the briefness of the video skim while holding adequate events for intra as well as interview in the multi-view environment.
- From the multimedia viewpoint, the proposed DELTA approach generates more accurate multi–view summary than [2, 8, 22, 37, 42], and a faster summary than [8, 22, 37].
- From multi-view VS with AdaBoosting has not been carried out to the best of our knowledge to generate the results. The proposed model still delivers a good summarization capability than the existing online as well as off-line summarization model.

2 Related work

In *VS*, preservation of the most interesting results without losing the semantics is referred as *Event Summarization (ES)*. For ES, Merler et al. [31] introduced a vectorized machine learning technique with 280 concept detectors to observe the significant issues. Due to low features, semantic gaps between important information are inadequate to demonstrate the high-level semantics [44]. Then, Alfaro et al. [1] suggested a novel approach to groups the keyframes in surveillance video, and then the basic temporal intra–class patterns. It is used to detect the video descriptor at the semantic, feature and priority level, which quantify the relevant local temporal similarities over the local activities [47]. Hidden Markov Model is applied to detect the cases in a Baseball game [5] while the important events are sparsely generated in a background of general [50]. In another novel work, Salehin et al. [33] summarize the mono-view surveillance videos using geometric primitives. Thus, a large number of VS models have introduced for video surveillance systems such as analyzing the social event [14, 29, 38, 46, 53] over the Internet. User scrutiny based VS is also suggested by Ma et al. [28], where several visual observations were made in the general context. Nevertheless, these observations seem time-consuming.

To achieve a fast summarization, Xu et al. [52] proposed a technique which considers the emotional and semantic clues as to the observations. An object is often detected in the

frame through the highly common area. During last decay, the event detection became a critical evidence for the sensing of the interestingness of multimedia content, due to the diversity and complexity of the surveillance videos. Moreover, CNN features are widely applied for event detection in surveillance and protection organizations. CNN features also attain good performance in image classification and action recognition tasks. In addition to this, the events are redefined after simultaneous detection of a set of frames based on the intrinsic property of CNN features on Deep Network (DevNet) [10]. A spatial-temporal saliency map is used for back, passing with DevNet, to extract the keyframes, and to secure the particular spatial position. Therefore, numerous models have been presented for VS over neural networks where computing resources are defined [3]. A deep CNN is also employed by Xu et al. [51] to find the events with sufficient leverage. However, the existing CNN's toolkits are used only at frame-level for motionless descriptors.

The traditional neural networks become impractically for the real-world image classification due to many causes. E.g., an image vectorization process will loose the complex 2D spatial structure; these networks are time-consuming from the computing perspective, etc. Suppose a 2D image with the size of 400 × 400 which needs 160,000 input nodes of a mesh. If the hidden layer comprises 80,000 nodes (i.e. 50% of the input nodes), the matrix should have 160,000 × 80,000 = 12,800,000,000 input weights. Moreover, this size will be more, for more number of layers, i.e., this number will be increasing even more rapidly. Therefore, a technique is required to sort out the actual-world images. The 2D convolution approach may be made instead of matrix multiplications. The 2D convolutions count the 2D structure of images. Moreover, a set of convolutional filters (size 11 × 11) also requires less time to learn than learning a large matrix (160,000 × 80,000) as indicated in Fig. 1. In Alex Net model [20] the first five layers are convolutional, and the final three are fully connected layers with 4096 filters.

For taking the above benefits, the authors decided to employ the AlexNet model [20] in the multi-view environment after one-month training on two GPUs with ILSVRC 2016 dataset1 of 1000 classes among 1.2 million training images. The principal applications of multi-view VS can consider in transport facilities patterns analysis, fast action recognition in the event of theft suspension, inspection of post-accidental circumstances, etc. Moreover, audiences and the interrelationships among multiple perspectives should be debated in the final summary; e.g. various events are concurrently recorded as a set of videos in security surveillance and monitoring organizations. The first multi-view summarization model is proposed by Fu et al. [8] based on the random walks graph. The correlations among various views are mapped with hypergraph, to achieve the momentous summary with minimum time. Subsequently, first online low complexity multi-view VS model [35] is introduced for saving transmission power and compression altogether keeping with the exciting content across wireless video sensors. Lately, a bipartite matching graph based multi-view VS model is brought out to produce the summaries of optimum path forest with clustering constraints [22].

3 Proposed model

In order to select the features, we first extract N frames in total with size $W \times H$ from a video $\mathbf{V}$, where W and H represents the width and height of a frame respectively. On an average, the video accessed from the data–sets consists 300 *RGB* frames. The various elements of the proposed DELTA model are presented in Fig. 2.

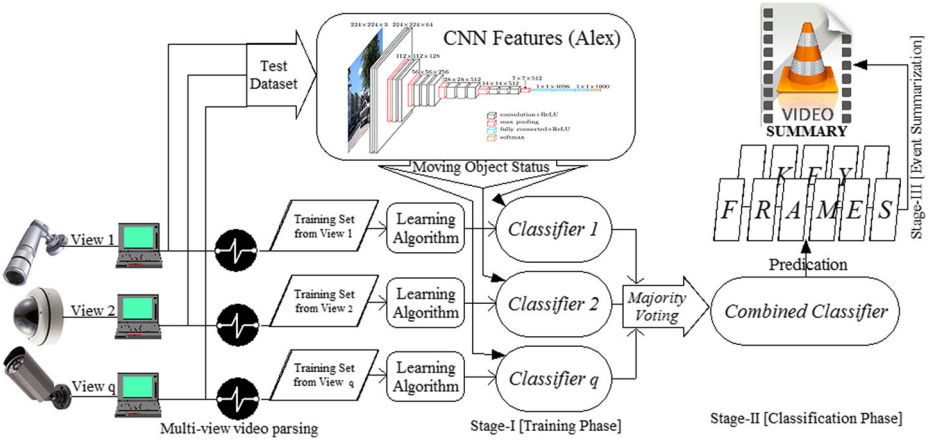

Fig. 2 Our proposed DELTA model with various components

3.1 Multi-view videos correlation with *AdaBoosting*

Most of the machine learning techniques often suffer from the curse of dimensionality, where a sample may have a lot of possible characteristics. For e.g.; 162,336 Haar features are used for object detection in the Viola-Jones framework [15, 19], where the image window size is 24 × 24 pixels. For evaluation, every feature may not entirely reduce the classifier training speed and implementation but also the prediction power. Although, a definitive meta-approach for machine learning ensemble works on a really powerful idea where sub-samples of the data are used for breeding. Unlike Support Vector Machine and neural networks, the *AdaBoosting* training policy chooses only the recognized features in order to enhance the prediction ability of the system. The irrelevant features do not need to be figured. Therefore, it improves the execution time while reducing the dimensionality as in analogy with video, frames are reduced at classification phase through AlexNet. It is primarily applied in statistical regression and classification. It too cuts down the divergence and helps us in avoiding the over-fitting through the removal of inactive/ moving object absence frames i.e. confusing samples [43]. Moreover, for over-fitting, initially, the authors fed only 10% of the frames instead 100% where each classifier are not for learning capacity of 100% frames. In addition to this, *AdaBooting* also has a great numeral of the applications in the field of computer vision and multimedia [12].

AdaBoosting often becomes effective, where employment is unstable in nonlinear models, i.e., a little variation in the training set sample gets the large changes in predictions. Similarly, the change may be observed frame by frame in a shot of a view of the video data set. It can be employed with any kind of method, where noise is the primary reason for the error in the learning of a model. Noise is an error which is created by the target function and variance comes from the sampling. The training samples can reduce the error from variance in the case of unstable classifiers or mis-classifiers, similarly as the frequent illumination changes in *multi-view* videos [49]. Thus, this technique starts with equally weighted data, then apply the first classifier to increase or decrease the weights on misclassified data, then apply the second classifier and so on. Then, majority vote policy is applied to the combined classifier for predicting the best keyframe.

Each classifier is targeting the individual view/ camera independently, while AlexNet is used to detect the moving an object in a frame by calculating the cosine similarity between two consecutive frames. Here, our objective is that a frame should be classified into active or inactive based on the moving objects (by AlexNet) and Support/ confidence of the classifier. These two factors are giving more accurate decision instead of only considering the object status in a frame (by AlexNet). Then, a combined classifier (in order to remove the statistical dependency among the multiple views) is used to generate the final output after taking the input from individually trained classifiers and the status of the object from AlexNet. If AlexNet is used for all the cameras individually, then the inter-view dependency among the views may not be resolved. Moreover, for multi-view perspective, in order to reduce the redundant frames in all the views and select only 1 keyframe out of q-frame at a time, then it may difficult to decide from which view, the keyframe will be selected where the object detection accuracy in a frame by AlexNet of individual q-views is same. In such circumstances, the individual classifier is used to make the decision whether the frame is selected or not by providing the additional information about the frame (support/ confidence parameter value for the frame) with object status in the frame. In the multi-view environment, for classifying the frames of the different cameras, AlexNet needs to be trained with respect to the individual view (the same trained AlexNet may not be used in different video/dataset because video data may be changed), which reduces the speed of the process. For considering the summarization problem in real-time applications, we trained the AlexNet once and used (only for detecting the object status, not for classification of the frames) in all types of videos/datasets and saved our training time for each view (if AlexNet need to train for each view), this helps to increase the speed of the summarization process. *AdaBoosting* implementation is discussed in Algorithm 1.

Training phase: let's say, a video dataset consists q views, i.e., q training samples are required as shown in Fig. 2. The training samples are chosen for every view by selecting the initial 10% frames of the views in order to ignore the training cost. The input is used from all individual view of a particular dataset, after drawing out the visual features through Alex model [20]. The AlexNet track moving object in a frame by calculating the cosine similarity between two consecutive frames. AlexNet is trained once in order to save the training time and used over entire video datasets, instead of training of AlexNet on individual views of a video dataset. It uses 1000 classes at its final or output layer. AlexNet is used on for detecting the moving objects, not for classification, Moving the object in a frame indicates that the frame may belong to an event which will be decided by the majority voting policy on q ensembles, otherwise frame will be discarded. Firstly, we trained these q classifiers as per the step 3 in algorithm 1. We will make the trained classifiers in D set along with a parameter α as a fraction of error which will be applied to calculate the support of an event in classifier phase as per the step 7 in the algorithm 1.

Adaboosting predictors are much potential to make the multiple versions (*multi–view* video) of a predictor and using them, to produce a combined predictor [21]. we A combined classifier $C_q(f)$ which can be interpreted as sticks with:

$$C_q(f) = \sum_{m=1}^{q} c_m(f) \tag{1}$$

Where c_m is a weak learner, which consume a frame f as input and output is either the positive value or negative value. Thus, the q^{th} classifier will be confident if the sample goes to the positive class and negative otherwise. At each iteration, the new weak learner is

selected and assigned a coefficient β_m, so that the sum of the training error E_m (see eq. 2) of the resultant m-stage boosting classifier should be minimal.

$$E_m = \sum_i E[C_{m-1}(f_i) + \beta_m O(f_i)] \tag{2}$$

Where, $C_{m-1}(f)$ is boosting classifier that has been made up at the old stage of training, and E (C) is some error function, and $c_m(f) = \beta_m O(f)$ is a weak learner where each weak learner generates an output, hypothesis O (f_I) on initial training samples, i represents a form number.

Algorithm 1 ES using modified AdaBoosting.

procedure *Event_AdaB*(VIEW $V_1, V_2, ..., V_q$)

Training phase: Stage-I

Initialize the parameters: • Initialize ensemble $D = \emptyset$.

• q=number of classifier (view of a video) to train.

• Set the weights $\mathbf{w} = [w_1, ..., w_N]$, $\sum_{j=1}^{N} w_j^1 = 1$, $w_j^1 \in [0, 1]$. (Usually, $w_j^1 = \frac{1}{N}$).

• **Z**=[$S_1, S_2..., S_q$.5 where $S_1 = 10\% \times length(V_1)$, $S_2 = 10\% \times length(V_2)$, ..., $S_q = 10\% \times length(V_q)$.

for l=1, ..., q **do**

• Take a sample S_l from **Z** using the distribution w^l.

• Build a classifier D_l using S_l as the training set.

• Calculate weighted ensemble error $\epsilon_l = \sum_{i=1}^{N} w_i^l k_l^i$

($k_l^i = 1$ if D_l mis-classifies z_i and $k_l^i = 0$ otherwise.)

if $\epsilon_l = 0$ or $\epsilon_l \geq 0.5$, **then** ignore D_l, reinitialize the weights w_i^l to $\frac{1}{N}$ and continue.

else, calculate $\alpha_l = \frac{\epsilon_l}{1-\epsilon_l}$, where $\epsilon_l \in (0, 0.5)$,

end if

Update the individual weights $w_i^{l+1} = \frac{w_i^l \alpha_l^{1-k_l^i}}{\sum_{j=1}^{N} w_j^l \alpha_l^{1-k_i^l}}$, where $i = 1, ..., N$.

end for

return D and $\alpha_1, ..., \alpha_q$

Classification phase: Stage-II

$E = \emptyset$; Number of Events

Calculate the support for an event E, among the moving object (detected by AlexNet) frames only.

$\mu_\gamma(X) = \sum_{D_l(X)=E_\tau} ln(\frac{1}{\alpha_l})$.

The event with the maximum support is chosen as the label for X as the keyframe of the event.

Event Summarization: Stage-III

Extracted Event Set $ES = \emptyset$.

Call(EBD($view_q$, S_{kl}, N, H, W)) for detecting the event boundaries in $view_q$

return ES : Events Summary from view $V_1, V_2, ..., V_q$

end procedure

Classification Phase: Secondly, the output of the trained ensembles is used as the input of a combined classifier [16, 43, 48], where *key frames* are predicted based on the majority vote policy (see stage-II in Fig. 2). E set is initiated as for finding the number of events. An event is selected which has the maximum support or confidence, the i.e. μ_γ value is higher of the frame X on q-trained classifiers and presence of moving the object by AlexNet. If there is no moving object in a frame, that particular frame will not go under compilation at trained classifiers which are saving our computing time. In another hand, if moving object status in a frame indicates the frame activeness, then the frame will undergo the trained classifiers which assign the support factor individually. If the support of a frame is the maximum out of the extracted frames in the event, the frame is declared as a key-frame of the event. A frame is selected only if any frame got 70% majority (estimated experimentally) from the classified trees where the frame holds moving object, otherwise not chosen and classifier tree will be pruning by filling the current node as leaf node [43] i.e. no moving object is a presence, so no activity. The final event summary is generated using the *key-frames* at stage-III in the Fig. 2.

Algorithm 2 Event boundaries of the event in $view_q$

procedure $EBD(view_q, S_{kf}, N, H, W)$
  S_{kf} selected *key–frames* from $view_q$.
  N: total frames in $view_q$.
  Event Boundary Threshold $[E_{BT}(\%)] = \frac{W \times H \times 90}{100}$
  Event Boundary Parameter: $[MIN, MAX] = [0, 0]$
  for each *key–frame* $kf \in S_{kf}$ **do**
    $j = k = find\_frame\_number\_in\_view_q(kf)$
    while $Eucli\_Dist(kf, k) \geq E_{BT}$ & $k \neq N$ **do**
      $k = k + 1$, where k is the current frame number.
    end while
    if $k == N$ **then** $MAX = N$, last frame of $view_q$.
    else $MAX = k$
    end if
    while $Eucli\_Dist(kf, j) \geq E_{BT}$ & $j \neq 1$ **do**
      $j = j - 1$, where j is the current frame number.
    end while
    if $j == 1$ **then** $MIN = 1$, first frame of $view_q$.
    else $MIN = j$
    end if
    Estimated boundaries of an event: $[MIN, MAX]$
  end for
end procedure

Event Summarization Phase: it hardly happens that an event consists single frame. Hence, we require defining the boundaries of the events. First, the frame number of a key-frame is identified from an appropriate view, after extracting the key-frames at stage-II. Second, in parliamentary procedure to discover the potential outcome, the boundary of an event (MIN: least frame number, MAX: maximum frame number) need to be well defined. The parameter MIN and MAX may or may not be equidistant from the key-figure number. Hence, the Event Boundary Threshold (E_{BT}) required estimating for an event boundary. Experimentally, $E_{BT}(\%) = \frac{H \times W \times 90}{100}$ is set based on the size (H: Height×

W: Width) of a frame as discussed at stage 4 in algorithm 2. A frame must have at-least $E_{BT}(\%)$ match with the keyframe, then only the frame will be counted in the event. The value of MAX (line 8 to 15) and MIN (line 16 to 23) are estimated as discussed in Algorithm 2. We iterate this algorithm q times to obtain all events in a particular dataset based on the extracted key-frames. Then, these events are arranged in temporal order for final summary.

4 Experiments and results

In order to evaluate, the proposed DELTA model, the qualitative, quantitative and computational analysis have been done in Sections 4.1, 4.2 and 4.3 respectively on *i7* processor and *16* GB of DDR4-RAM desktop computer. Three datasets[1,2] (*Lobby, Office, BL-7F*) with total 26 *multi-view* videos are accessed from [35] with the following details:

Dataset	# views	Duration (s)	Camera Issues
Lobby	3	1482	Not fixed & Crowded
Office	4	3016	Not Synchronized
BL-7F	19	8170	Overlapped field view

- *Lobby (3 views)* [35] was recorded by synchronizing fixed and unstable surveillance systems. The scenes are more crowded with enriching activities and become more challenging for summarization.
- *Office (4 views)* [35] was captured by fixed and unstable surveillance cameras. Asynchronism, different frame rates and frequent illumination changes, create more difficulty for a good video summarization.
- *BL-7F (19 views)* [35] was synchronized views with fixed cameras from 7^{th} floor of Barry Lam Building, captured the office suite and entire hallway. High density and many overlapped fields of views create an obstacle to summarize the multi-perspectives.

For observing the moving objects through visual CNN features and cosine similarity between two consecutive frames, AlexNet_7 (see Fig. 3) The network is trained with seven convolutional layers, close to 0.19 Billions parameters and 1000 categories. The accuracy, number of layers and parameters (in Billion) of different version of the AlexNet [20] are shown below:

Model	# Layers	Parameters	Accuracy (%)
AlexNet [20]	5	1.97 Bn	78.67
AlexNet_4	4	2.10 Bn	76.02
AlexNet_6	6	0.26 Bn	81.35
AlexNet_7	7	0.19 Bn	**83.57**

The utility of AlexNet assists us to remove the inactive/ no motion frame by appending the moving object status with every frame during the testing phase. The classification is done on the active frames based on the majority voting policy on the trained classifiers. In the multi-view environment, if AlexNet is used for classifying the frames of the different

[1] http://media.ee.ntu.edu.tw/research/summarization/

[2] https://cs.nju.edu.cn/ywguo/summarization.html

```
Using TensorFlow backend.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 96, 310, 230)      34944
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 96, 155, 115)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 256, 151, 111)     614656
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 256, 75, 55)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 384, 73, 53)       885120
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 384, 71, 51)       1327488
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 256, 69, 49)       884992
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 256, 34, 24)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 128, 32, 22)       295040
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 128, 28, 18)       409728
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 128, 14, 9)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 16128)             0
_________________________________________________________________
dense_1 (Dense)              (None, 9216)              148644864
_________________________________________________________________
dense_2 (Dense)              (None, 4096)              37752832
_________________________________________________________________
dense_3 (Dense)              (None, 1000)              4097000
=================================================================
Total params: 194,946,664
Trainable params: 194,946,664
Non-trainable params: 0
```

Fig. 3 AlexNet_7 : 7 Convo layers AlexNet [20]

cameras, we need to be trained the AlexNet with respect to the individual view (same trained AlexNet may not be used in different video dataset because video data may be changed), which reduces the speed of the process and inter-view dependency among the views may not be resolved. We trained the AlexNet once and used (only for detecting the object status, not for classification of the frames) in all types of videos/ datasets and saved our training time. If DELTA model is not using the AlexNet, then the computation time (after executing all the frames) definitely will more than the current setup which may not fit into the real-time computation.

4.1 Qualitative analysis

All three (*Lobby, Office, BL-7F*) *multi-view* datasets of the surveillance videos along with their ground truths are accessed from [8, 22] for several tests. These ground truth summaries have been taken in the judgment. Here, the central frame or the center nearest frames of an event are referred as a *keyframe*.

The keyframes extracted from the individual perspective of office dataset which is ordered in the temporal order in Fig. 4. The proposed model detected the 20 events (20

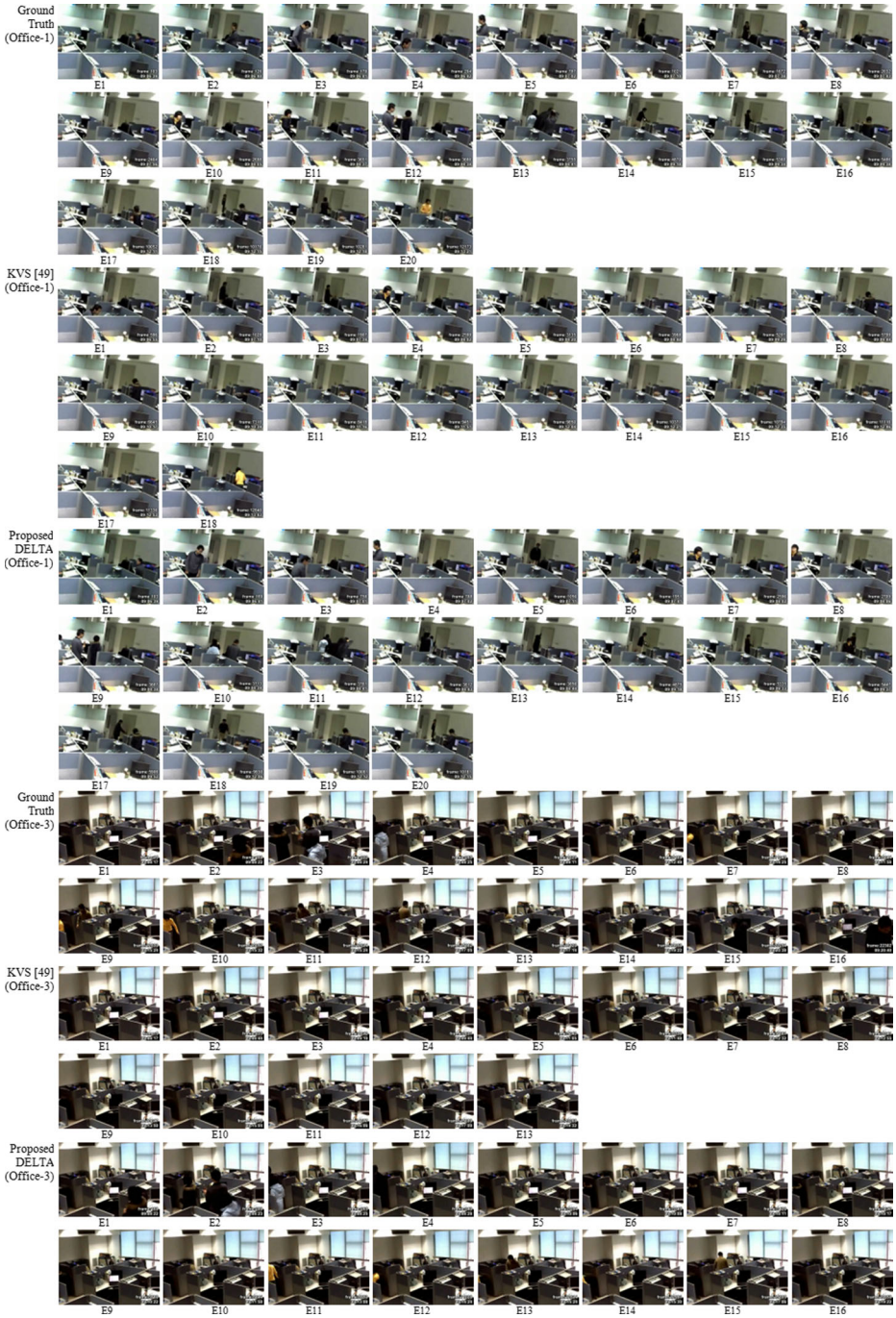

Fig. 4 Representative frames ($Ei : i^{th}$ event) of view-1, and view-3 from *Office* dataset arranged in temporal order

events [37]) from view-0, 20 events (18 events [37]) from view-1, and (13 events [37]) 16 events from view-3 of the Office dataset for mono-view video summarization. For multi-view perspective, Fig. 5 show the extracted representative frames of the observed events are set up in the temporal order which is detected by Bipartite Graph approach [22] and by the proposed DELTA model respectively. Here, we remarked that the representative frames (20) of detecting events (20 events [37]) out of 25 (decided as ground truth) are discovered by

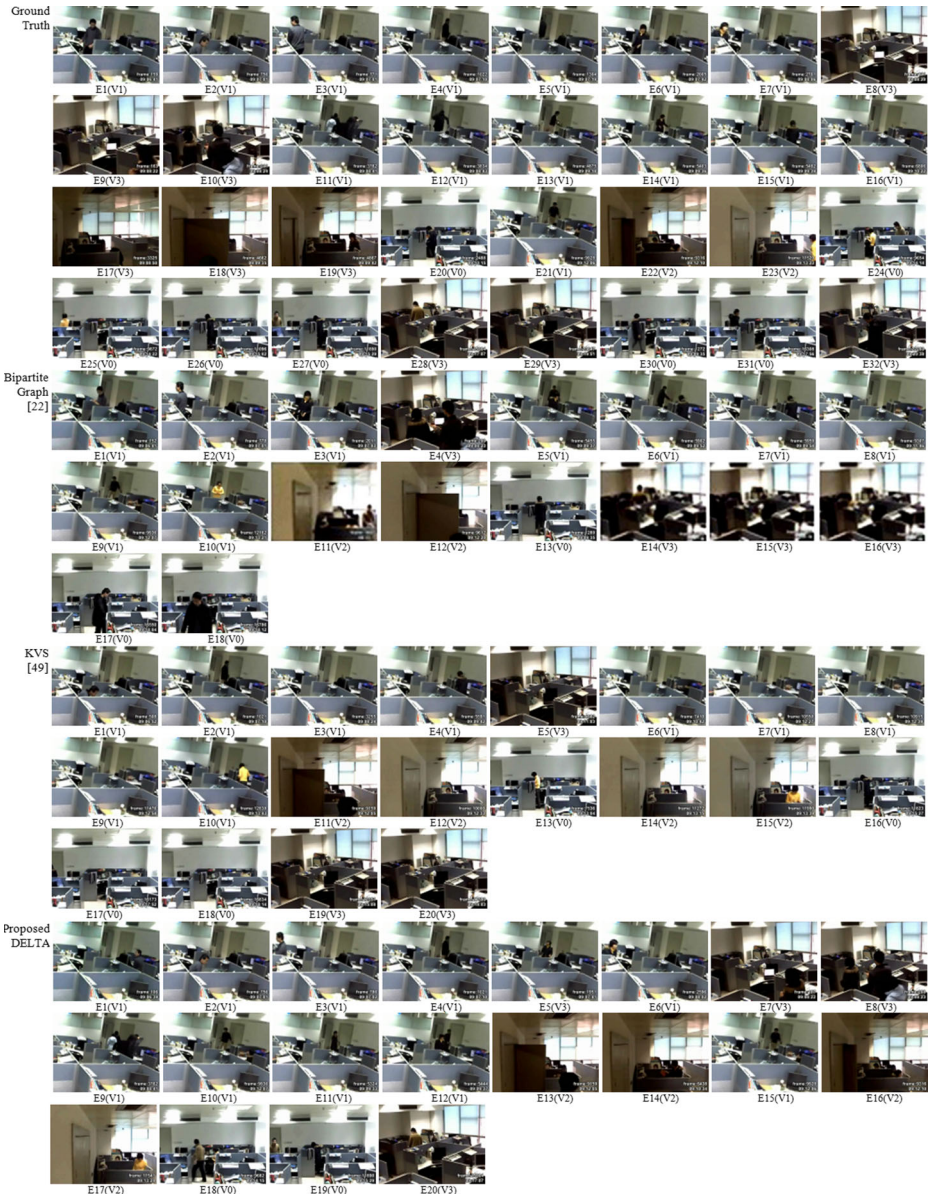

Fig. 5 Multi-view keyframes ($Ei(Vj)$: j^{th} View of i^{th} Event;) on Office dataset, extracted are arranged in temporal order

our proposed DELTA model, are significantly heavier than the representative figures (18) of detected events by Bipartite Graph approach [22]. Figure 6 proves the extra keyframes extracted by our approach which is not evoked by other existing models. Hence, our proposed DELTA approach is able to extract the maximum number of events as per the ground truth in the mono-view environment, while in the multi-view environment; our approach extracted more number of the events than the most of the state-of-the-art models. Hence, DELTA model quantitatively outperforms than existing techniques.

4.2 Quantitative analysis

In order, to attain the best summary of the surveillance systems, *keyframes* extraction as well video skimming techniques have been extensively discussed in *multi–view* summarization. However, There is no standard approach which can appraise their public presentation. To an extent, the character of each summarization model, summaries of video can be developed from several approaches are equated to the ground summaries. Their measurement is based on the below three assessment metrics, those contain commonly *Precision, Recall* and *F–measure*. *Precision* and *Recall* is estimated by considering the marked ground truth in a frame unit by the following instructions:

- *True Positive (TP):* a frame that is chosen by an algorithm which connects with an interesting activity.
- *False Positive (FP):* a frame that is chosen by an algorithm but does not associate with an interesting activity.
- *False Negative (FN):* Any frame that is not picked out by an algorithm, but associates with an interesting activity.

The definition of *Precision, Recall* and *F-measure* are delineated in equation 3, equation 4 and equation 5 respectively:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F - measure(F_\rho) = \frac{(1 + \rho^2) \times Precision \times Recall}{\rho^2 \times Precision + Recall} \tag{5}$$

The higher value of *Recall* represents a bigger fraction of interesting forms are returned by the overture. High *Precision* represents the turn of interesting frames, which are returned

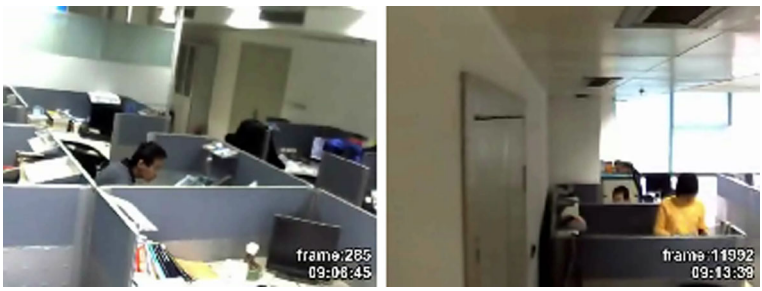

Fig. 6 Extra Extracted *key–frames* from view 1 (2^{nd} event) and view 2 (17^{th} event) by our approach as shown in Fig. 5

more as recognized from a number of uninteresting returned frames. The users often counted the equal priorities to both *Precision* and *Recall* matrix in order to assess the *VS* approach as much as accurate. For $\rho = 1$, F_1 become the Harmonic mean of *Precision* and *Recall*. It is too known as balanced *F–score* due to the equal weight of the *Recall* and *Precision*. Therefore, high $F_1 - measure$ means more accurate summarization model.

We equated our results with the ground summaries. The parameter BETA $\beta = \frac{H \times W}{d}$ is defined using H (height of a frame), W (width of a frame) and d (10 experimentally estimated). The value of parameter BETA is used to estimate the similarity (*Euclidean* distance) between an extracted *key–frame* of an event and the ground truth frames. If the similarity between such two is greater than or equal to β then only, we consider that the frame as a matched frame in the calculation of *Precision* and *Recall* for the quantitative analysis. We compared the experimental results with the results which was provided by [2, 22, 37, 42] proposed the more complex *VS* models that produces good *mono–view* video summaries.

Tables 1, 2, 3 and 4 show the quantitative evaluation of the suggested approach for *mono–view* summarization with state–of–art models with $\rho = 1$, where the best answers are expressed in bold and it is remarked that

- *Extracted events* by our suggested model for *mono–view* summarization on all three data sets are maximum among all other models.
- *Summary length* is indicate the length of the summarized video should always be lesser than the actual video.
- *Precision* of our DELTA model for *mono–view* summarization on all three data sets is maximum except *Office-2*, *BL–7*, *BL–9* and *BL–10* video among all other models. Thus, DELTA is robust to eliminate the low active frames.
- The sequence BL-13 of the BL-7F dataset is used to indicate that if a picture has no consequences at all, and so a good example should not produce the summary, i.e. no frame should be taken. Hence, we can test the model using BL-13 (no event) whether

Table 1 *Mono–view* comparison on *Lobby* dataset

View	Algorithm	Summary Length (s)	# Events Detected	Precision (%)	Recall (%)	F–measure (%)
	Tree Based (D=30) [42]	145/**494**	14/**32**	74.5	49.7	60.0
	Tree Based (D=90) [42]	**094**/**494**	08/**32**	83.7	36.2	51.0
Lobby–0	Compressed Domain [2]	249/**494**	13/**32**	43.6	50.0	47.0
	KVS [37]	225/**494**	**32**/**32**	65.1	**91.7**	76.1
	Proposed DELTA	119/**494**	**32**/**32**	**92.7**	78.2	**84.8**
	Tree Based (D=30) [42]	177/**494**	21/**33**	75.7	57.9	66.0
	Tree Based (D=90) [42]	**108**/**494**	12/**33**	84.9	39.7	54.0
Lobby–1	Compressed Domain [2]	247/**494**	18/**33**	46.9	50.2	48.5
	KVS [37]	198/**494**	30/**33**	79.4	**88.3**	83.6
	Proposed DELTA	158/**494**	**31**/**33**	**94.0**	78.0	**85.3**
	Tree Based (D=30) [42]	157/**494**	20/**32**	78.4	52.1	63.0
	Tree Based (D=90) [42]	**089**/**494**	08/**32**	77.7	29.2	43.0
Lobby–2	Compressed Domain [2]	248/**494**	17/**32**	48.1	50.4	49.2
	KVS [37]	206/**494**	29/**32**	77.6	85.8	81.5
	Proposed DELTA	178/**494**	**32**/**32**	**91.3**	**87.9**	**89.6**

Table 2 *Mono–view* comparison on *Office* dataset

View	Algorithm	Summary Length (s)	# Events Detected	Precision (%)	Recall (%)	F–measure (%)
	Tree Based (D=30) [42]	396/**895**	19/**25**	15.7	**80.7**	26.0
	Tree Based (D=90) [42]	259/**895**	19/**25**	21.6	72.4	33.0
Office–0	Compressed Domain [2]	165/**895**	11/**25**	22.5	48.2	31.0
	KVS [37]	154/**895**	**20/25**	88.1	49.1	63.1
	Proposed DELTA	**072/895**	**20/25**	**99.5**	49.1	**65.7**
	Tree Based (D=30) [42]	175/**544**	15/**20**	31.8	**70.0**	44.0
	Tree Based (D=90) [42]	091/**544**	08/**20**	44.8	51.3	48.0
Office–1	Compressed Domain [2]	233/**544**	10/**20**	17.3	50.6	26.0
	KVS [37]	068/**895**	18/**20**	**97.2**	65.2	**78.0**
	Proposed DELTA	**067/544**	**20/20**	96.5	63.1	76.3
	Tree Based (D=30) [42]	275/**682**	19/**21**	16.3	**80.1**	27.0
	Tree Based (D=90) [42]	150/**682**	16/**21**	24.0	64.2	35.0
Office–2	Compressed Domain [2]	227/**682**	07/**21**	11.9	48.1	19.0
	KVS [37]	157/**895**	**21/21**	70.1	60.6	**65.0**
	Proposed DELTA	**073/682**	**21/21**	**95.6**	37.9	54.3
	Tree Based (D=30) [42]	477/**895**	**16/16**	05.5	**92.5**	10.0
	Tree Based (D=90) [42]	387/**895**	14/**16**	06.4	87.2	12.0
Office–3	Compressed Domain [2]	232/**895**	09/**16**	05.8	47.3	10.0
	KVS [37]	087/**895**	13/**16**	78.0	42.1	54.7
	Proposed DELTA	**056/895**	**16/16**	**97.4**	37.5	**55.2**

the model is generating a correct summary or not. With this, the proposed model solves one of the significant challenges that even if the video has no event it must not detect any. In Table 4, "-" denotes there is an absence of events. The most of the existing models failed to detect the absentia of event.

- *Recall* of the proposed DELTA model on all three data sets is better than the most of the models.
- Some of the models are better in terms of *Precision* while others in terms of *Recall*. Hence, *F–measure* which is the harmonic mean of *Precision* and *Recall* is a more honest bar. *F–measure* of our model on all three datasets except *Office-2*, *Office-3*, *BL–0*, *BL–15* and *BL–18* video is also maximum among all the other models. It stands for the better execution of our model than existing models.

We compared the experimental results with the results that were provided by [8, 22] which introduced more complex *VS* models, generate good *multi–view* video summaries. The authors noted that lots of redundancy due to the simultaneous occurrence of most of the events in video summaries which are picked out from all the *mono–view* models. Therefore, *multi–view* summarization is urgently needed for working video surveillance applications such as behavior analysis, event detection and summarization and activity detection, anomaly detection [3]. Table 5 presents the quantitative evaluations of the DELTA model to *multi–view* summarization with state–of–the –art where the best answers are expressed in bold. Here, we observed for the multi-view environment that

Table 3 *Mono–view* comparison on *BL–7F* dataset

View	Algorithm	Summary Length (s)	# Events Detected	Precision (%)	Recall (%)	F–measure (%)
	Tree Based (D=30) [42]	217/**430**	7/**8**	09.9	**91.1**	18.0
BL–0	Compressed Domain [2]	208/**430**	2/**8**	05.3	46.8	10.0
	KVS [37]	274/**430**	6/**8**	78.5	90.3	**84.0**
	Proposed DELTA	**028/430**	**8/8**	**94.7**	74.3	83.3
	Tree Based (D=30) [42]	274/**430**	**7/7**	08.2	**83.8**	15.0
BL–1	Compressed Domain [2]	**037/430**	2/**7**	28.6	38.9	33.0
	KVS [37]	182/**430**	6/**7**	48.2	81.8	60.7
	Proposed DELTA	052/**430**	**7/7**	**89.6**	51.8	**65.6**
	Tree Based (D=30) [42]	164/**430**	12/**20**	44.6	56.6	50.0
BL–2	Compressed Domain [2]	215/**430**	07/**20**	29.1	48.5	36.0
	KVS [37]	164/**430**	17/**20**	74.7	**72.7**	73.7
	Proposed DELTA	**091/430**	**19/20**	**87.2**	67.7	**76.1**
	Tree Based (D=30) [42]	206/**430**	**2/2**	02.2	**100**	04.0
BL–3	Compressed Domain [2]	005/**430**	**2/2**	61.6	68.4	65.0
	KVS [37]	093/**430**	**2/2**	58.3	66.8	62.3
	Proposed DELTA	**004/430**	**2/2**	**78.5**	56.0	**65.4**
	Tree Based (D=30) [42]	153/**430**	11/**13**	30.2	77.6	43.0
BL–4	Compressed Domain [2]	050/**430**	04/**13**	53.5	45.0	49.0

- *Recognized events* by the DELTA model for *multi-view* summarization on all three data sets is maximum among all other models.
- *Recall* of the DELTA model to *multi–view VS* on two (*Lobby, Office*) datasets are maximum among all other existing techniques. It indicates that the suggested access is robust to preserve the interesting content as events.
- *Precision* of our model in *multi–view* summarization on a high density and overlapped views dataset *BL-7F* is maximum among all other models.
- *F–measure* of the proposed DELTA model to *multi–view* summarization on all three surveillance videos data–set are also maximum among all the other existing systems which show better functioning of our proposed DELTA surveillance video summarization model.

4.3 Computational complexity

For experiment perspective, we employed a video shot of 300 frames (i.e. 10 seconds) by frame size 352×240 from online *VS* [35]. We implemented DELTA technique in order to bring down the computational price. We noticed that on an average, DELTA technique takes less than 90 seconds including the preprocessing time during the training phase where training samples are selected by 10% of the initial video content from all views instead of the entire views of the *Office* dataset at stage-I. At stage-II, it requires less than 300 minutes for all four views of *Office* dataset for classifying/ testing whether the figure selected as *keyframe* or not. The proposed DELTA model requires total processing time altogether less than 390 seconds to complete as compared with the computational cost of 600 seconds which is reported by Kuanar et al. [22].

Table 4 *Mono–view* comparison on *BL–7F* dataset

View	Algorithm	Summary Length (s)	# Events Detected	Precision (%)	Recall (%)	F–measure (%)
BL–4	KVS [37]	267/**430**	**12/13**	**83.4**	**80.3**	**81.8**
	Proposed DELTA	**047/430**	11/**13**	**83.4**	**80.3**	**81.8**
	Tree Based (D=30) [42]	274/**430**	**6/6**	05.8	**93.9**	11.0
BL–5	Compressed Domain [2]	042/**430**	4/**6**	19.5	48.3	28.0
	KVS [37]	073/**430**	5/**6**	65.4	70.5	67.9
	Proposed DELTA	**035/430**	**6/6**	**88.3**	50.4	**64.2**
	Tree Based (D=30) [42]	334/**430**	**1/1**	00.1	**100**	00.0
BL–6	Compressed Domain [2]	**002/430**	0/**1**	00.0	00.0	00.0
	KVS [37]	013/**430**	**1/1**	22.2	**100**	36.3
	Proposed DELTA	003/**430**	**1/1**	**42.4**	**100**	**59.6**
	Tree Based (D=30) [42]	327/**430**	**3/3**	08.3	67.9	15.0
BL–7	Compressed Domain [2]	020/**430**	0/**3**	**91.7**	45.7	61.0
	KVS [37]	027/**430**	2/**3**	87.7	**80.4**	84.0
	Proposed DELTA	**016/430**	**3/3**	91.5	78.5	**84.5**
	Tree Based (D=30) [42]	334/**430**	**2/2**	02.0	**100**	04.0
BL–8	Compressed Domain [2]	**003/430**	0/**2**	50.0	25.1	33.0
	KVS [37]	031/**430**	**2/2**	67.1	**100**	80.3
	Proposed DELTA	**003/430**	**2/2**	**86.2**	**100**	**92.6**
	Tree Based (D=30) [42]	353/**430**	1/**2**	09.0	49.6	15.0
BL–9	Compressed Domain [2]	020/**430**	0/**2**	**91.7**	28.8	44.0
	KVS [37]	069/**430**	**2/2**	63.1	**90.5**	74.4
	Proposed DELTA	**019/430**	**2/2**	88.9	86.8	**87.9**
	Tree Based (D=30) [42]	252/**430**	1/**3**	18.7	54.1	28.0
BL–10	Compressed Domain [2]	028/**430**	0/**3**	**94.1**	30.7	46.0
	KVS [37]	093/**430**	2/3	76.5	**90.1**	82.7
	Proposed DELTA	**026/430**	**3/3**	86.4	82.9	**85.7**
	Tree Based (D=30) [42]	135/**430**	05/**13**	56.3	28.0	37.0
BL–11	Compressed Domain [2]	163/**430**	03/**13**	62.5	37.6	47.0
	KVS [37]	065/**430**	10/**13**	89.5	**97.6**	93.4
	Proposed DELTA	**045/430**	**11/13**	**98.8**	96.5	**97.7**
	Tree Based (D=30) [42]	222/**430**	14/**16**	44.1	64.8	53.0
BL–12	Compressed Domain [2]	215/**430**	06/**16**	35.2	50.1	41.0
	KVS [37]	222/**430**	**15/16**	89.6	**93.1**	91.3
	Proposed DELTA	**121/430**	**15/16**	**100**	87.4	**93.3**
	Tree Based (D=30) [42]	346/**430**	**0/0**	–	–	–
BL–13	Compressed Domain [2]	213/**430**	**0/0**	–	–	–
	KVS [37]	**000/430**	**0/0**	–	–	–
	Proposed DELTA	**000/430**	**0/0**	–	–	–
	Tree Based (D=30) [42]	156/**430**	09/**10**	24.8	67.6	36.0
BL–14	Compressed Domain [2]	**022/430**	01/**10**	78.5	30.7	44.0
	KVS [37]	073/**430**	09/**10**	70.1	**88.2**	78.1

Table 4 (continued)

View	Algorithm	Summary Length (s)	# Events Detected	Precision (%)	Recall (%)	F–measure (%)
	Proposed DELTA	043/**430**	**10/10**	**94.2**	84.2	**88.9**
	Tree Based (D=30) [42]	157/**430**	4/**9**	25.4	60.4	36.0
BL–15	Compressed Domain [2]	**023/430**	1/**9**	91.0	32.2	48.0
	KVS [37]	048/**430**	**9/9**	**95.3**	**96.4**	**95.8**
	Proposed DELTA	039/**430**	**9/9**	**95.3**	95.0	95.1
	Tree Based (D=30) [42]	154/**430**	**14/14**	32.6	**84.8**	47.0
BL–16	Compressed Domain [2]	045/**430**	08/**14**	64.2	48.8	55.0
	KVS [37]	064/**430**	**12/14**	81.0	83.6	82.3
	Proposed DELTA	**042/430**	**14/14**	**86.2**	83.6	**84.9**
	Tree Based (D=30) [42]	167/**430**	11/**14**	42.4	68.4	52.0
BL–17	Compressed Domain [2]	210/**430**	08/**14**	24.2	49.1	32.0
	KVS [37]	075/**430**	**13/14**	93.5	**88.0**	90.7
	Proposed DELTA	**090/430**	**13/14**	**97.4**	86.9	**91.9**
	Tree Based (D=30) [42]	165/**430**	**4/4**	08.5	**90.7**	16.0
BL–18	Compressed Domain [2]	022/**430**	0/**4**	29.1	40.8	34.0
	KVS [37]	056/**430**	3/**4**	66.0	70.5	**68.2**
	Proposed DELTA	**019/430**	**4/4**	**79.3**	54.9	64.8

The execution time of the temporal ordering of the extracted events by the proposed approach at stage-III is about the same time as reported by Kuanar et al. [22] in OPF clustering. Therefore, we can say that the proposed DELTA model is almost 33% faster than Bipartite Graph approach [22]. Thus, the proposed DELTA model can meet the requirement of real-time applications, such as video surveillance and protection organizations, sports highlights, etc. Average computation time per video over a particular dataset is compared in Table 6.

Table 5 A multi-view comparison of the proposed DELTA model with with state–of–art

Dataset	Algorithm	Summary Length (s)	# Events Detected	Precision (%)	Recall (%)	F–measure (%)
	Bipartite Graph [22]	0176/**1482**	33/**35**	**100**	76.7	86.8
Lobby	Fu et al. [8]	0158/**1482**	**34/35**	**100**	79.0	88.3
	KVS [37]	0389/**1482**	33/**35**	83.6	87.5	85.5
	Proposed DELTA	**0155/1482**	**34/35**	84.9	**93.1**	**88.8**
	Bipartite Graph [22]	**0059/3016**	18/**25**	**100**	69.2	81.8
Office	KVS [37]	0208/**3016**	20/**25**	76.8	91.9	83.7
	Proposed DELTA	0080/**3016**	**20/25**	81.4	**92.9**	**86.7**
	Bipartite Graph [22]	0633/**8170**	N/A	75.9	**98.2**	85.0
BL–7F	KVS [37]	0571/**8170**	43/**51**	81.6	74.9	78.1
	Proposed DELTA	**0302/8170**	**46/51**	**99.6**	76.3	**86.4**

Table 6 Computational time comparison

Algorithm	Sampling Rate [frame per Sec]	Average Time per video [Sec]
Fu et al. [8]	25–30	225.00
Bipartite Graph [22]	30	150.00
KVS [37]	30	138.63
Proposed DELTA	30	**097.25**

5 Conclusion

Adaboosting machine learning approach is used in order to get the accurate decision for active frames and still forms. We first trained our ensembles using a meta approach, where views interdependency and illumination changes are considered in the preparation stage. We used the CNN features in order to achieve the best summarization by selecting the potential keyframes. Then, the test set is processed through training ensemble to generate the keyframes of the event. At the end, we detect the event boundary using these *key–frames* for video skimming. Moreover, the model does not need prior knowledge about the number of issues in a video. Experiments showed that our video summarization DELTA model successfully reduced the video data while keeping the interesting scene in the course of results. Our model outperformed the state–of–the–art models on all three datasets with the best *F–measure*. The DELTA approach extracted the upper limit number of events of the all three data sets. The computing analysis of our proposed DELTA model also showed that it can fulfill the requirement of real–time applications [9, 18, 24].

References

1. Alfaro et al (2016) Action recognition in video using sparse coding and relative features. CVPR, 2688–2697
2. Almeida J et al (2013) Online video summarization on compressed domain. JVCIR 24(6):729–738
3. Anurag K et al (2017) A novel superpixel based color spatial feature for salient object detection. IEEE CICT'17
4. Brunelli R et al (1999) A survey on the automatic indexing of video data. JVCIR 10(2):78–112
5. Chang P et al (2002) Extract highlights from baseball game video with hidden markov models. IEEE ICIP 1:I-609
6. Chang et al (2016) They are not equally reliable: semantic event search using differentiated concept classifiers. CVPR, 1884–1893
7. Fu Y et al (2010) Multi view video summmarization. IEEE TMM 12(7):717–729
8. Fu Y et al (2014) Multi-view metric learning for multi-video video summarization, CoRR, vol. abs/1405.6434, [Online]. Available: http://arxiv.org/abs/1405.6434
9. Gagandeep S et al (2017) PICS: a novel technique for video summarization. Springer MISP'17
10. Gan et al (2015) Devnet: adeep event network for multimedia event detection and evidence recounting. CVPR, 2568–2577
11. Gygli et al (2015) Video summarization by learning submodular mixtures of objectives. CVPR, 3090–3098
12. Hao W et al (2006) Generalized multiclass adaboost and its applications to multimedia classification. CVPR'06 Workshop, 113–113
13. Jasim H et al (2016) Surveillance video summarization based on histogram differencing and sum conditional variance. WASET Inter J Comp Elect Automat Control Inform Engg 10(9):1652–1657
14. Jiang G et al (2015) Super fast event recognition in internet videos. IEEE TMM 17(8):1174–1186
15. Jones et al (2006) Method and system for object detection in digital images. U.S. Patent No. 7,099,510.29

16. Krishan K et al (2017) F-DES: fast and deep event summarization. IEEE TMM
17. Krishan K et al (2017) SOMES: an efficient SOM technique for event summarization in multi-view surveillance videos. Springer ICACNI'17
18. Krishan K et al (2017) V-LESS: a video from linear event summarieS. Springer CVIP'17
19. Krishan K et al (2017) D-CAD: deep and crowded anomaly detection. ACM ICCCT'17
20. Krizhevsky A et al (2012) Imagenet classification with deep convolutional neural networks. ANIPS, 1097–1105
21. Krogh A et al (1995) Neural network ensembles, cross validation, and active learning. ANIPS 7:231–238
22. Kuanar S et al (2015) Multi-view video summarization using bipartite matching constrained optimum-path forest clustering. IEEE TMM 17(8):1166–1173
23. Kumar K et al (2016) Equal partition based clustering approach for event summarization in videos. SITIS, 119–126
24. Kumar K et al (2017) Key-lectures: keyframes extraction in video lectures. Springer MISP'17
25. Kumar K et al (2017) Eratosthenes sieve based key-frame extraction technique for event summarization in videos. MTAP, 1–22
26. Kumar K et al (2017) Event BAGGING: a novel event summarization approach in multi-view surveillance videos IEEE IESC'17
27. Lu S et al (2014) A bag-of-importance model with locality-constrained coding based feature learning for video summarization. IEEE TMM 16(6):1497–1509
28. Ma F et al (2005) A generic framework of user attention model and its application in video summarization. IEEE TMM 7(5):907–919
29. Mazloom M et al (2014) Conceptlets: selective semantics for classifying video events. IEEE TMM 16(8):2214–2228
30. Mazloom M et al (2016) TagBook: a semantic video representation without supervision for event detection. IEEE TMM 18(7):1378–1388
31. Merler M et al (2012) Semantic model vectors for complex video event recognition. IEEE TMM 14(1):88–101
32. Mundur P et al (2006) Keyframe-based video summarization using Delaunay clustering. IJDL 6(2):219–232
33. Musfequs S et al (2016) Video summarization using geometric primitives. IEEE DICTA'16
34. Nagasaka A (1991) Automatic video indexing and full-video search for object appearances. In: Conf. on visual database system, pp 119–133
35. Ou H et al (2015) On-line multi-view video summarization for wireless video sensor network. IEEE J S T Sig Process 9(1):165–179
36. Panagiotakis C et al (2009) Equivalent key frames selection based on iso-content principles. TCSVT 19(3):447–6451
37. Potapov D et al (2014) Category-specific video summarization. ECCV, 540–555
38. Qian S et al (2016) Multi-modal event topic model for social event analysis. IEEE TMM 18(2):233–246
39. Singh N et al (2016) A convex hull approach in conjunction with Gaussian mixture model for salient object detection. DSP, 22–31
40. Singh N et al (2016) A novel position prior using fusion of rule of thirds and image center for salient object detection MTAP. https://doi.org/10.1007/s11042-016-3676-8
41. Sun X et al (2000) Video summarization using R-sequences. Real-Time Imag 6(6):449–459
42. Valdes V et al (2008) Binary tree based on-line video summarization. ACM TRECVid video summarization workshop, 134–138
43. Vezhnevets A et al (2007) Avoiding boosting overfitting by removing confusing samples. Machine learning: ECML, 430–441
44. Wang M et al (2012) Event driven web video summarization by tag localization and key-shot identification. IEEE TMM 14(4):975–985
45. Wang S et al (2014) Semi-supervised multiple feature analysis for action recognition. IEEE TMM 16(2):289–298
46. Wang F et al (2014) Video event detection using motion relativity and feature selection. IEEE TMM 16(5):1303–1315
47. Wang et al (2015) Video event recognition with deep hierarchical context model. CVPR, 4418–4427
48. Weber B (2008) Generic object detection using AdaBoost. Department of Computer Science University of California, Santa Cruz
49. Wu B et al (2004) Fast rotation invariant multi-view face detection based on real adaboost. IEEE FGR'04, 79–84
50. Xiong Z et al (2004) Effective and efficient sports highlight extraction using the minimum description length criterion in selecting GMM structures. IEEE ICME'04 3:1947–1950

51. Xu et al (2015) A discriminative CNN video representation for event detection. CVPR, 1798–1807
52. Xu B et al (2016) Fast summarization of user-generated videos: exploiting semantic, emotional, & quality clues. IEEE TMM 23,3:23–33
53. Yang X et al (2015) Automatic visual concept learning for social event understanding. IEEE TMM 17(3):346–358
54. Zhang T et al (2012) A generic framework for video annotation via semi-supervised learning. IEEE TMM 14(4):1206–1219

Krishan Kumar is working as an Assistant Professor in National Institute of Technology, Uttarakand, India. He obtained M. Tech (Computer Science and Engineering) from Visvesvaraya National Institute of Technology, Nagpur, India in 2014. Presently, he is pursuing Ph.D. (Computer Science and Engineering) from Visvesvaraya National Institute of Technology, Nagpur, India. His current research areas are: Virtualization, Cloud computing, Computer Vision, Deep learning, image processing, object detection, pattern recognition, video analysis and summarization.

Dr. (Mrs) Deepti D. Shrimankar received the Ph.D. degree in Parallel Computing from Visvesvaraya National Institute of Technology, Nagpur, India, in 2013. The B. Tech. degree in Computer Technology and M.Tech. degree in image processing from Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, Maharastra, India, in 1997 and 2007 respectively. She is working as an Assistant Professor with Visvesvaraya National Institute of Technology, Nagpur. She is currently supervising six Ph.D. candidates and working on a project of "Genome-wide identification of cis-regulatory elements and development of G-logo tool for genome logo representation" with "Young Scientist Research Fellowship Grant" by DietY for the academic session 2014-15. Her research interests include computer networks, parallel computing, image processing, video analysis and summarization, wireless sensor networks and cryptography.