Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder

Dinesh Singh, *Student Member, IEEE*, and Chalavadi Krishna Mohan, *Member, IEEE*

***Abstract*— Vision-based detection of road accidents using traffic surveillance video is a highly desirable but challenging task. In this paper, we propose a novel framework for automatic detection of road accidents in surveillance videos. The proposed framework automatically learns feature representation from the spatiotemporal volumes of raw pixel intensity instead of traditional hand-crafted features. We consider the accident of the vehicles as an unusual incident. The proposed framework extracts deep representation using denoising autoencoders trained over the normal traffic videos. The possibility of an accident is determined based on the reconstruction error and the likelihood of the deep representation. For the likelihood of the deep representation, an unsupervised model is trained using one class support vector machine. Also, the intersection points of the vehicle's trajectories are used to reduce the false alarm rate and increase the reliability of the overall system. We evaluated out proposed approach on real accident videos collected from the CCTV surveillance network of Hyderabad City in India. The experiments on these real accident videos demonstrate the efficacy of the proposed approach.**

***Index Terms*— Accident detection, anomaly detection, deep learning, stacked autoencoder.**

I. Introduction

SMART cities are using various innovative technologies to improve the peoples quality of life [1], [2]. European Commission in 2010 [3] has launched one such highly visible and important initiative named *European Initiative on Smart Cities*. In smart cities sustainable transportation is a critical dimension where the goal is to build 1) intelligent public transportation systems based on real-time information, 2) traffic management systems for congestion avoidance, and 3) safety and green applications [4]. However, the growing size of cities and increasing population mobility have determined a rapid increase in the number of vehicles on the roads, which has resulted in many challenges for road traffic management authorities among them road accidents require immediate attention to reduce the loss of life and properties. Traffic accidents caused an estimated 1.2 million deaths in 2004, with 50 million people injured [5]. Due to various security concerns, all the main cities across the world already installed a significant number of cameras for traffic monitoring purpose. The use of these already existing surveillance camera networks will be a viable solution, but these systems mostly rely on human observation. For human observers, it is almost impossible to monitor and recognize unusual events without missing in such a large number of camera scenes in real time [6]. Thus, it raises the need for automated solutions for accident detection.

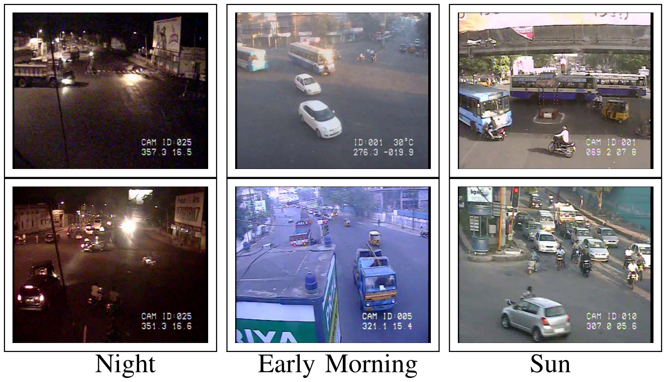


Fig. 1. The sample video frames, showing various difficulties in the collected video dataset for accident detection. The major challenges are the low visibility in the night videos, poor quality of videos, traffic congestion, occlusions, etc.

Over the recent years, researchers from both industry and academia have been working to develop automatic detection methods using computer vision and pattern recognition techniques, but the level of current technology is still limited to apply them in the real world. Devising vision-based algorithm for this task is very challenging. In practice, the performance of computer vision based traffic accident detection algorithms can be challenged by many factors [7]–[9]. These factors include imaging conditions (varying illumination and changing weather conditions), environments (urban, highways), as shown in Fig. 1.

As also pointed out by Yun *et al.* [10], the existing methods for traffic accident detection developed till date can be categorized into three approaches:

- **Modeling of traffic flow patterns:** In this category, the typical law-full traffic patterns (namely, go-straight, U-turn, right-turn, etc.) are modeled as baseline [11], [12] and any deviation from this model is considered as an abnormal traffic event. This approach will work only when the normal traffic pattern appears at

Manuscript received July 8, 2017; revised January 23, 2018 and April 11, 2018; accepted May 6, 2018. Date of publication May 28, 2018; date of current version February 28, 2019. The Associate Editor for this paper was H. Huang. *(Corresponding author: Dinesh Singh.)*

The authors are with the Visual Learning and Intelligence Group, Department of Computer Science and Engineering, IIT Hyderabad, Hyderabad 502205, India (e-mail: cs14resch11003@iith.ac.in; ckm@iith.ac.in).

Digital Object Identifier 10.1109/TITS.2018.2835308

a fixed region repeatedly, thus unable to detect collisions which are essential to accident detection.

- **Analysis of vehicle activities:** The methods in this categories first detect the moving vehicles and then extract motion features such as the distance between two vehicles, acceleration, direction, etc. of a vehicle from moving vehicles' tracks [12]–[18]. However, unsatisfactory tracking performance in crowded traffic scenes becomes their bottleneck and limits their usage.
- **Modeling of vehicle interactions:** These methods have been inspired by sociological concepts and model the interaction among vehicles and detect accidents [19], [20]. However, a large number of training data and use of speed change information alone limit the performance of these methods.

Deep learning techniques have recently been applied to various computer vision tasks such as image classification [21], [22], semantic segmentation [23], [24], object detection [25], human action/activity recognition [26], visual traking [27]–[30] etc. Such great success of deep learning techniques is mostly attributed to their outstanding performance in representing visual data. In this work, we exploit the deep learning for accident detection. We introduce an unsupervised deep learning framework to automatically learn discriminative features for accident detection in the surveillance videos. We propose a new approach to learning appearance and motion features as well as their correlations similar to what Xu and Ricci [31] used for anomaly detection in pedestrian. Deep learning methods for combining multiple modalities have been investigated in previous works [32], [33]. However, to our knowledge, this is the first work where multimodal deep learning is applied for accident detection. A double fusion scheme is proposed to combine appearance and motion features for discovering abnormal activities [34]. However, these techniques are not applied for accident detection till date. The proposed method is validated on challenging real accident videos and the results obtained are very motivating.

The rest of the paper is organized as follows. Section II presents related work. We present the proposed approach in Section III. Section IV discusses the experimental setup and results. We conclude in Section V with future directions.

II. Related Work

This section presents the state-of-the-art for accident detection and enlightens the pros and cons of these methods.

Ki and Lee [18] detects accidents by setting a predefined threshold on the certain parameters such as position, acceleration, and direction of vehicles. Moving objects are obtained by taking the difference between two successive frames. A similar approach also presented by Hui *et al.* [13] where the parameters are computed from the trajectories of the moving vehicles obtained through background modeling using Gaussian mixture model (GMM) and tracking using mean shift algorithm. This method is simple and easy to implement and deploy but not suitable in unconstrained environments like a frequent change in traffic pattern and weather conditions since it relays only on the change in position and speed parameters. Also, the dependence only on the change in speed and position can easily lead to false alarms like a sudden movement of a vehicle.

Aköz and Karsligil [16] obtained moving vehicles using moving blob detection and tracking using Kanade-Lucas-Tomasi (KLT) Tracker. The trajectories of various vehicles for normal traffic are clustered using continuous hidden Markov model (C-HMM) to set up a baseline of normal traffic which requires a large amount of training data to acquire all possible activity paths. If a given unseen trajectory shows significant deviation from baseline trajectories, then an accident is declared. Although, this method does not rely on speed parameters but the whole trajectory of a vehicle. Sadek *et al.* [15] use histogram of flow gradient to obtain the orientation of flows. From the optical flow or velocity obtained, the Euclidean distances between the centers of gravity of patterns are calculated, and then logistic regression is used to predict the probability of the occurrence of an accident. But, this method uses logistic regression, the interpretation of which is merely a probability.

The tracking of vehicles is a key component in the accident detection but tracking in the dense traffic and abrupt motion is a challenging problem because the scenario typically contains abrupt changes in the appearance and motion of the target. A significant research is carried out in tracking under abrupt motion. Kwon and Lee [35] proposed a robust tracking method by alleviating the motion smoothness constraint in abrupt motion using Wang-Landau Monte Carlo (WLMC) sampling method in tracking algorithm. Lim *et al.* [36] handle the tracking under abrupt motion by applying optimised swarm-based sampling strategy for proposal selection. Su *et al.* [37] used visual saliency model integrated with a particle filter for recovering lost track due to abrupt motion by detecting the target region from salient regions obtained from the saliency map of current frame.

These existing methods make use of motion or track of moving objects and simply try to define a normal baseline (many time using pre-decided threshold only) and any unknown event not obeying this baseline is simply declared as an accident. Although, the deviations in motion parameters gives useful pre-collision information but do not sufficient for accident detection.

III. Proposed Framework for Accident Detection

The course of accident can be divided into three stages: pre-collision, collision, and post-collision. Each stage gives us a significant amount of information but also involves several difficulties as discussed below.

Pre-Collision: The pre-collision case is the most vital information to explain an accident scenario. Also, this information may become a good evidence for crime scene investigation. The pre-collision situation is a clear violation of traffic rules by any/both the vehicles, which include violation of traffic lane, violation of signals at intersections, violation of speed limit at congested roads, abrupt motion on the road, etc. Finally, we can say that pre-collision stage is an unusual activity and thus can be easily detected by applying anomaly [31], [38]

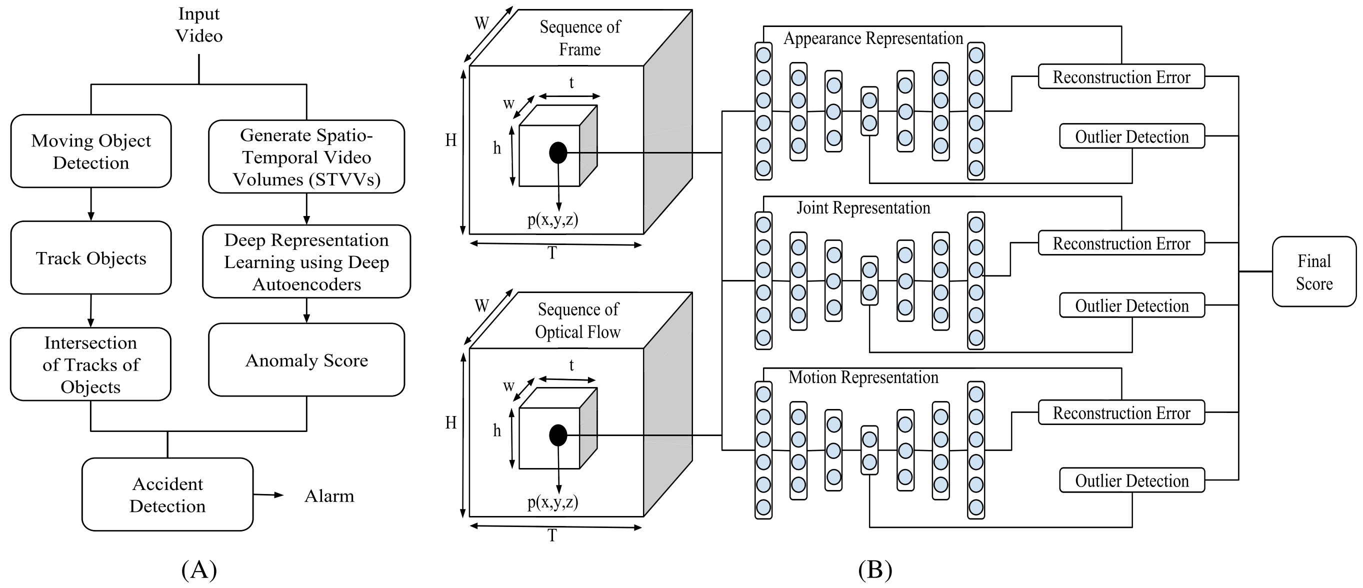


Fig. 2. The architecture of the proposed framework for accident detection. (A) Overview of the framework. It consists of two streams, one for the generation of the collision score using the trajectories of the moving vehicles and the other one for generation of abnormality score using deep representation. (B) A detailed diagram of abnormality detection using deep stacked autoencoder on three modalities, namely, appearance, motion, and joint representation.

detection methods based on the various parameters, such as speed, trajectories, position, etc.

Collision: The collisions are essential to accident detection, but it is very complicated to detect and cannot be directly detectable by any general purpose computer vision technique. One way to detect a collision is to identify the joints of the trajectories of the vehicles over spatiotemporal dimensions. However, the major challenge is the discrimination between collision and occlusion. For this we use the trajectories over space-time interest points [39] and improved dense trajectories [40], [41].

Post-collision: As stated above that the collision and occlusions are hard to classify and may lead to false alarms. These false alarms further can be refined by considering the post-collision scene. The two most common post-collision scenes include: 1) Fallen objects at the collision point: As we stated that the intersection of the trajectories of two vehicles might be a collision or an occlusion. However, after the intersection, if both the trajectories are continued, and no abrupt or zig-zag motion resulted. Then, the intersection is merely an occlusion, not a collision. However, if some abrupt motion or discontinued trajectories have occurred, then the possibility of a collision is high. Measure the time for which the object remains static. 2) Crowd attention towards the collision point: The last and final stage of the accident is the crowded road or pedestrians running towards the collision point.

As shown in Fig. 2 the proposed framework for automatic detection of accident incident composed of abnormality detection using the deep representation of spatiotemporal video volumes (STVVs) and collision detection using intersection points of trajectories. The anomaly detection works in two steps, the first step is the automatic training of the deep features and the second step is to determine the outlier score for unknown incidents. The separately stacked denoising autoencoder (SDAE) trained over STVVs from the previously seen normal traffic video one for each representation is used to generate the deep representation for the STVVs from the unseen traffic video. The possibility of an accident is determined based on the reconstruction error and the likelihood of the deep representations for which outlier score is generated using one-class support vector machine (SVM). All these individual scores (a.k.a. local score) are then fused to compute the final decision to declare an incident as an accident. We present the detail description of these steps in the following subsections.

A. Spatio-Temporal Volume Generation

In order to localize the accident incident, we divided the entire video into several smaller size volumes called spatiotemporal video volumes (STVVs) similar to [26], with different scales in both space and time as well as across the modalities such as appearance, motion, and joint representations. Fig. 3 shows a STVV at a pixel $p(x, y, z)$ in a 3D video volume.

Lets, $\mathbf{v} \in \mathbb{R}^{W\times H\times T}$ given continuous video sequence where point $\mathbf{v}(x, y, z) \in \mathbb{R}$ gives the intensity of the pixel (x, y, z) for all $x \in [0, W]$, $y \in [0, H]$, and $z \in [0, T]$. Here, $\mathbf{v}(0 : W,\ 0 : H,\ z)$ represents the z^{th} frame. The $\mathbf{v}(x - \frac{w-1}{2} : x + \frac{w-1}{2}, y-\frac{h-1}{2} : y+\frac{h-1}{2}, z-\frac{t-1}{2} : z+\frac{t-1}{2})$ is a space-time video volume (STVV) of size $w \times h \times t$ around the pixel (x, y, z). These STVVs are then normalized and vectorized into a vector $\mathbf{x} \in \mathbb{R}^{wht}$. Finally, we have a datasets $\mathbf{X} = \{\mathbf{x}_i\}, i = 1, 2, \cdots, n$ where n is total number of such STVVs.

B. Stacked Denoising Autoencoder (SDAE)

A denoising autoencoder (DAE) is a simple one-hidden-layer neural network with unsupervised learning using back-propagation algorithm. The objective of a DAE is to transform

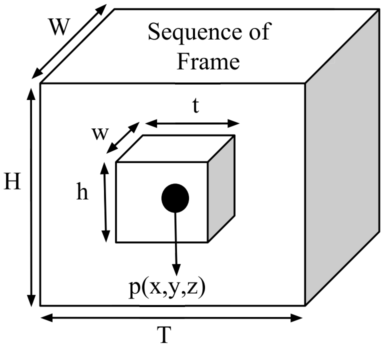


Fig. 3. The generation of the spatio-temporal video volumes (STVVs). The STVVs are the pixels in the immediate vicinity of a point $p(x, y, z)$ covered by a 3D sliding window of size (w, h, t).

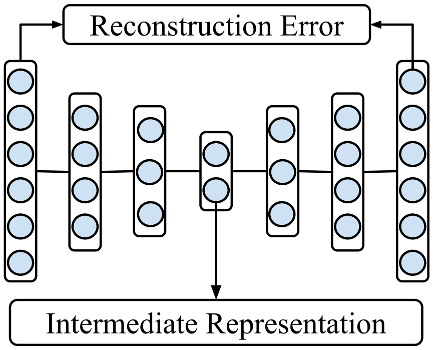


Fig. 4. The network topology of the proposed stacked autoencoder used to model the baseline for the normal traffic. The network consists three decoder layers followed by three decoder layers. The reconstruction error is the Euclidean distance of the input and output layers. The output of the middle layers is the latent intermediate representation.

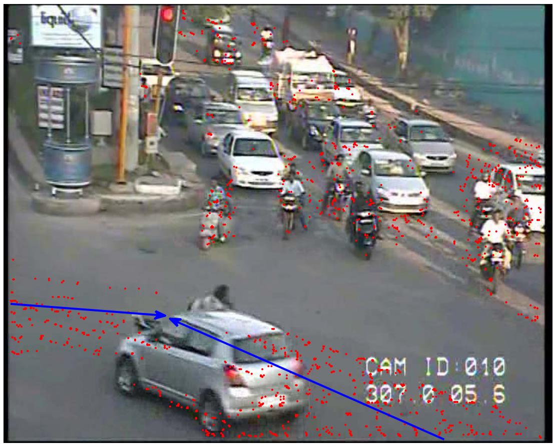


Fig. 5. The intersection of two trajectories during an accident. The trajectories of the motorcycle and car intersect each other also there is no further progress in the trajectories of the motorcycle and car, thus considered as a collision.

given partially corrupted samples into a compressed representation to learn latent patterns by minimizing the amount of distortion in reconstructed samples. The denoising autoencoder consists of two processes:

1) *Encoding:* The encoder takes a nonlinear mapping denoted as $f_e(\mathbf{x}_i|\mathbf{W}, \mathbf{b})$ from the partially corrupted input to a hidden representation. For a given corrupted input $\tilde{\mathbf{x}}_i$, a compressed hidden layer representation h_i can be obtained as below:

$$\mathbf{h}_i = f_e(\tilde{\mathbf{x}}_i|\mathbf{W}, \mathbf{b}) = \sigma(\mathbf{W}\tilde{\mathbf{x}}_i + \mathbf{b}). \tag{1}$$

Typically, corrupted inputs are obtained by drawing samples from a conditional distribution $p(\mathbf{x}|\tilde{\mathbf{x}})$, for example the Gaussian white noise or salt-pepper noise.

2) *Decoding:* The decoder is used to map the hidden representation back to a reconstruction representation through a similar transformation $f_d(\mathbf{h}_i|\mathbf{W}', \mathbf{b}')$. For a given hidden representation $\mathbf{h}_i$, a reconstructed representation $\hat{\mathbf{x}}$ is computed as below:

$$\hat{\mathbf{x}}_i = f_d(\mathbf{h}_i|\mathbf{W}', \mathbf{b}') = s(\mathbf{W}'\mathbf{h}_i + \mathbf{b}'). \tag{2}$$

Here, $< \mathbf{W}, \mathbf{b} >$, and $< \mathbf{W}', \mathbf{b}' >$ denote the weights and the bias terms of the encoder and decoder, respectively. The $\sigma(\cdot)$ and $s(\cdot)$ are activation functions. Typically, the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ is used as the activation function. The network can learn a more stable and robust representations of the input using this encoder/decoder structure. An stacked denoising autoencoder (SDAE) is a cascade of several denoising autoencoders (DAEs) as shown in Fig. 4.

The parameters $(\mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}')$ are learned for a given training set $X = \{\mathbf{x}_i\}_{i=1}^n$ by minimizing the following regularized least square optimization problem:

$$\min_{\mathbf{W},\mathbf{W}',\mathbf{b},\mathbf{b}'} \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \lambda(\|\mathbf{W}\|_F^2 + \|\mathbf{W}'\|_F^2), \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The first term $\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$ is the average reconstruction error, while the second term $(\|\mathbf{W}\|_F^2 + \|\mathbf{W}'\|_F^2)$ is the weight penalty for regularization. The importance of these two terms is balanced by parameter λ. Typically, sparsity constraints are also imposed on the output of the hidden units to discover meaningful representations from the data.

C. Detection of Intersection Points in Trajectories

First, we detect moving objects by subtracting background images, and then the moving objects are tracked. In an STVV, if two tracks are intersecting each other then it represents either a collision or an occlusion as shown in Fig. 5. In the presented frame, we found that the trajectories of the bike and car intersect each other. Also, the trajectories of several other vehicles touch each other several time. Since the trajectories continue in the subsequent frames, they are simply considered as the occlusions, not collisions. But, there is no further progress in the trajectories of the bike and car so this is considered as a collision. The collision scores $\mathcal{C}$ of a STVV is the simple count of such points in that STVV.

D. Accident Score Generation

We use one-class SVM to generate the outlier score γ of intermediate representation $\mathbf{h}$ for a given STVV. One-class SVM requires only one class data and fits an outer boundary around this data. In this case, we use only STVVs from normal traffic for building model. The outlier score γ for a given $\mathbf{h}$ is computed from one-class SVM as below:

$$\gamma = f(\mathbf{h}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{h}_i, \mathbf{h}) - \rho, \tag{4}$$

where, $\{\mathbf{h}_i, \cdots, \mathbf{h}_m\}$ are the m support vectors with their respective Lagrange multipliers α_i, ρ is the threshold value. If the weighted density of a feature vector with support vectors is above a threshold ρ then feature vector is classified as normal and abnormal otherwise. The values of these parameters are computed by solving below dual problem for n training points $\{\mathbf{h}_i, \cdots, \mathbf{h}_n\}$:

$$\max_{\bar{\alpha}} \ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{h}_i, \mathbf{h}_j)$$
$$subject\ to \sum_{i=1}^{n} \alpha_i = 1 \ and \ 0 \le \alpha_i \le \frac{1}{\upsilon n}, \tag{5}$$

where $\upsilon \in (0, 1)$ control the penalty imposed on the nonzero slack variables.

For a STVV $\mathbf{x}$ the reconstruction error ξ is computed as below

$$\xi = \|\mathbf{x} - \hat{\mathbf{x}}\|_{\mathcal{F}}^2. \tag{6}$$

where, $\hat{\mathbf{x}}$ is the reconstruction of the STVV $\mathbf{x}$ and $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm. The high reconstruction error shows that the particular STVV is less likely drawn from the previously seen patches and thus increases the likelihood that it may belong to an accident scene.

For each STVV $\mathbf{v}$, we extract three representation: (i) appearance representation $\mathbf{x}^A$ based on still frames, (ii) motion representation $\mathbf{x}^M$ based on optical flow, and (iii) joint representation $\mathbf{x}^J$ by early fusion (concatenation) of both appearance and motion representation. For each representation, we extract deep representation using stacked denoising auto-encoder and compute anomaly scores $\gamma^A, \gamma^M, \gamma^J$ using Equation (4) and reconstruction errors ξ^A, ξ^M, ξ^J using Equation (6). Also, the collision score $\mathcal{C}$ is computed as discussed in previous section. Finally, we use post-fusion of scores to get single final score. We consider the linear combination to keep less number of parameters and reduced computation in comparison to a non-linear combination. The computation of non-linear transformation leads to a large number of parameters and increased computation time. The final accident score s is given as below:

$$s = \beta_1 \gamma^A + \beta_2 \gamma^M + \beta_3 \gamma^J + \beta_4 \xi^A + \beta_5 \xi^M + \beta_6 \xi^J + \beta_7 \mathcal{C}, \tag{7}$$

where, $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$, and, β_7 are free parameters to control false alarms. The final decision of whether $\mathbf{v}$ corresponds to an accident or not is taken based on the threshold s_T which is given as.

$$Decision = \begin{cases} Accident, & \text{for } s > s_T \\ Normal, & \text{otherwise.} \end{cases} \tag{8}$$

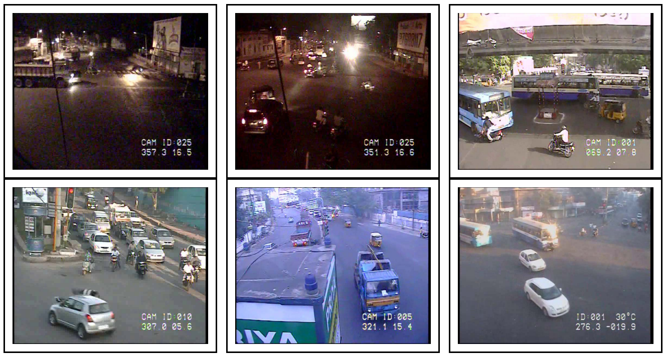


Fig. 6. Sample frames from the video dataset used to evaluate the performance of proposed approach. The dataset contains videos of accidents during the various environmental conditions such as high sunlight, night, early morning as well as from different cameras and view angles.

The parameters in Equation (7) are computed using linear regression on a small amount of manually labeled data as follows. Let, $\mathbf{X}$ be the set of STVVs with corresponding label set $\mathbf{y}$, where $y_i = \{-1, +1\}$, and $\mathbf{S} = [\boldsymbol{\gamma}^A, \boldsymbol{\gamma}^J, \boldsymbol{\gamma}^M, \boldsymbol{\xi}^A, \boldsymbol{\xi}^J, \boldsymbol{\xi}^M, \boldsymbol{\mathcal{C}}]$ be the set of corresponding scores. Then the parameters set $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7]^T$ is given by

$$\boldsymbol{\beta} = (\mathbf{S}^T \mathbf{S} + \lambda \mathbf{I})^{-1} \mathbf{S}^T \mathbf{y}, \tag{9}$$

where $\lambda = 10^{-6}$ is the regularization parameter. While the best performing threshold s_T is decided empirically.

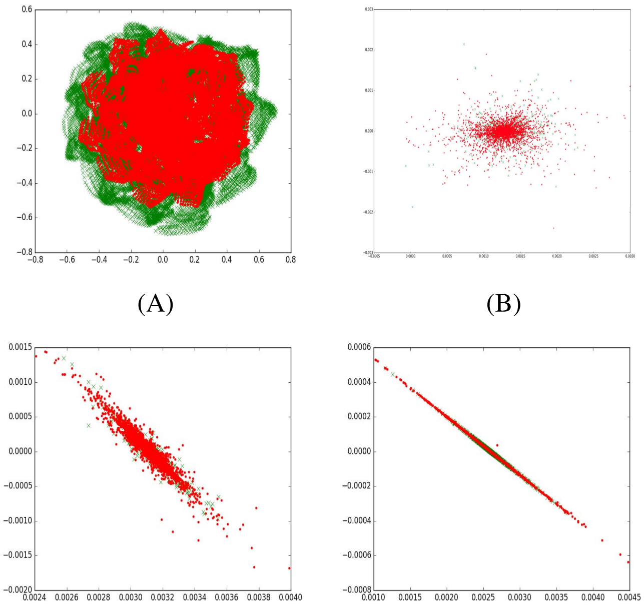
IV. Experimental Evaluation

The experiments are conducted on a machine running Ubuntu 16.04 Xenial Xerus having specification Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz×48 processor, 128GB RAM with NVIDIA Corporation GK110GL [Tesla K20c]×2 GPUs. Programs for feature representation are written in $Python - 2.7.11$, where for video processing we use $OpenCV - 2.4.9$, for implementing auto-encoder we use $Keras - 1.1.1$, for one class SVM we use $fitcsvm$ function of MATLAB.

A. Dataset Used

Since there is no public video dataset available for accident detection, we collected own dataset from the CCTV surveillance network of Hyderabad City in India. Video clips collected from City surveillance network are captured at 30 frames per second. Fig. 6 presents samples from the collected dataset. Each video clip starts few minutes before the incident of an accident and contains several minutes after the incident. First few minutes of video which contains normal situation are used for training the model and remaining for testing. There are total 127138 normal frames, and 863 frames contain partial or full accidents labeled manually. For training 94720 normal frames are used. For testing we used 33280 frames 32417 normal and 863 accident frames. The dataset is made public for the research community for further comparison[1].

[1] https://sites.google.com/site/dineshsinghindian/iith_accident-dataset



(A)

(B)

(C)

(D)

Fig. 7. The 2-D visualization of the distribution of the STVVs data generated from a sample video. The STVVs during normal and accident are shown using the green cross and red dot, respectively. [Best viewed in color]. (a) Input. (b) Layer-1. (c) Layer-2. (d) Layer-3.

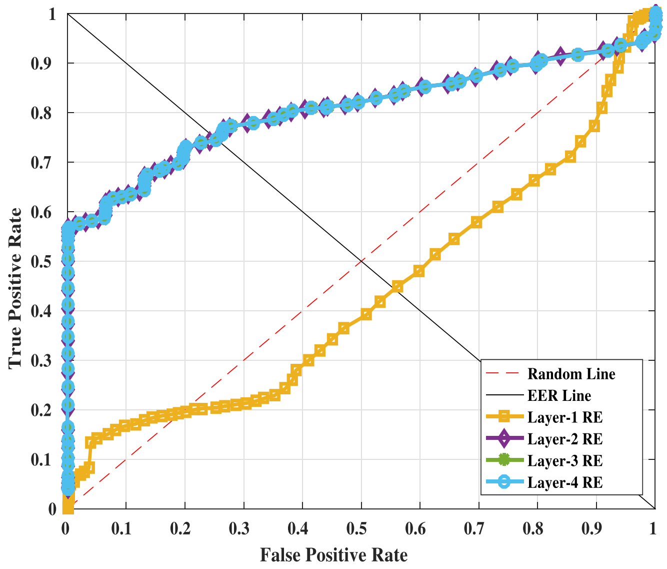


Fig. 8. ROC curve for accident detection using reconstruction error at various Layers. [Best viewed in color].

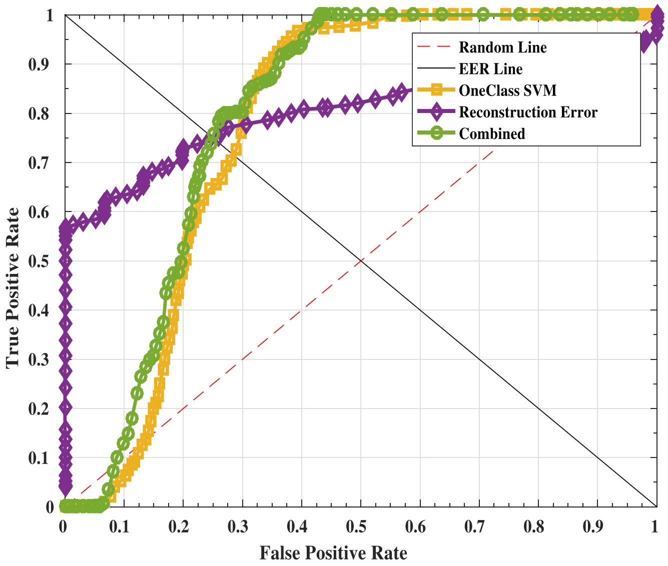


Fig. 9. ROC curve for accident detection using reconstruction error (RE), one class SVM, and their combination. [Best viewed in color].

B. Results and Discussion

The STVVs are generated at various scales in both space and time. For experiment we generate STVVs of spatial scale of 11×11, 13×13, and 15×15 pixels. For each spatial scale, we generate three temporal scales of 3, 5, and 7 frames. Thus finally we generate 9 STVVs at each spatiotemporal point.

The denoising stacked autoencoder projects high dimensional data onto a lower dimension where it forms a manifold as shown in Fig. 7. The projections of the unknown patterns which are drawn from the similar class patterns from which the SDAE is trained are very close to the other points in the manifold and very far otherwise. Thus the one-class-SVM with RBF kernel generates the score based on how likely an unknown pattern is drawn from the normal traffic patterns. A high deviation score confirms that the particular pattern belongs to some unusual/unseen/abnormal/outlier event, which further increases the possibility of an accident as an accident is also a rare event.

Since the proposed method is an unsupervised method and the final classification is based on a threshold. Changing the threshold results into a change in the performance. Thus to find the optimal threshold we computed various performance scores on hundred different threshold values. The trade-off between the sensitivity (i.e., true positive rate) and the specificity (i.e., false positive rate) is shown via ROC curve. In a ROC curve, the red dotted line shows the random prediction (50%) line, and the solid black line is the equal error rate (EER) line. We analyzed the discrimination ability for all three representations for reconstruction error at various layers of SAE as well as the outlier score of the corresponding intermediate representations using one-class-SVM. Fig. 8 illustrates the ROC curve for the experiments conducted for various thresholds (s_T) for reconstruction error at different layers of the stacked denoising autoencoder. The area under the curve (AUC) increases with an increase in the number of stacked autoencoders. But after stack of two autoencoders, there is a very slight increment, and thus the performance (AUC) for them is not changing significantly than the performance of layer-2. The final performance (AUC) of the accident detection based on the reconstruction error alone are 76.54%, 51.57%, and 76.28 for appearance, motion, and joint representations, respectively.

A similar phenomenon is also seen for outlier score using one-class SVM on the intermediate representation. The performance increases with an increase in the number of autoencoders. The final performance (AUC) of the accident detection based on the intermediate representation using one-class SVM is 77.54%, 62.87%, and 74.21% for appearance, motion, and joint representations, respectively. Fig. 9 illustrate the ROC

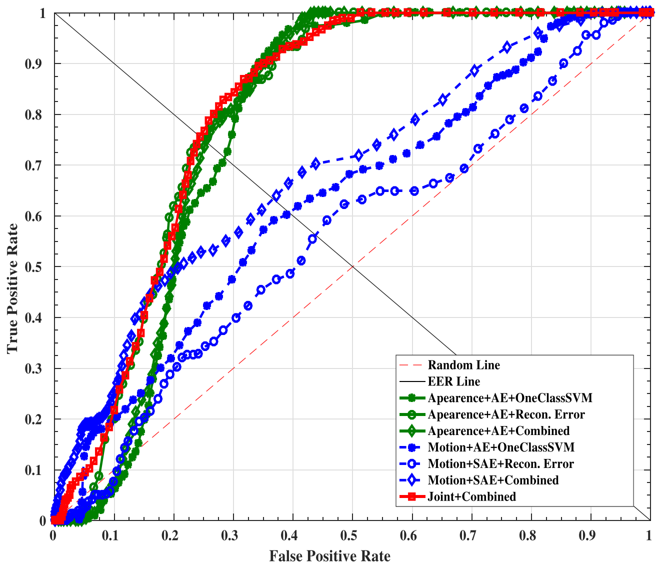


Fig. 10. ROC curve for accident detection using reconstruction error, one class SVM, and their combination for the appearance, motion, and their joints representations. [Best viewed in color].

TABLE I
AREA UNDER CURVE (AUC) FOR VARIOUS MODALITIES AND METHODS

Representation	**Recon. Error**	**OneClassSVM**	**Combined**
Appearance Representation	0.7654	0.7754	0.7760
Motion Representation	0.5157	0.6287	0.6291
Joint Representation	0.7628	0.7421	**0.8106**

curve for the experiments conducted for various threshold s_T for reconstruction error, one-class-SVM, and their combination for the stack of three auto-encoders. The AUC for the combined is more than both reconstruction error and one-class-SVM alone.

However, the performance increases when we combine both the scores. Fig. 10 illustrates the ROC curve for the experiments conducted for various threshold s_T for reconstruction error, one-class-SVM, and their combination for the appearance, motion, and their joints representations. The AUC using different modalities and methods is listed in Table I. The AUC for the combined is more than both reconstruction error and one-class-SVM alone. The final performance of the accident detection based on the combined representation is 77.60%, 62.91%, and 81.06% for appearance, motion, and joint representations, respectively.

Fig. 11 show the examples of the detected accident regions using the anomaly scores from various samples from the collected dataset. The red region is the predicted accident regions using a single score either by appearance, motion, or intersection of tracks while the green box shows the region decided using final score. Here, the accident detection rate using anomaly score is very high as it accurately detects almost all the regions which are declared accident manually (i.e. ground-truth). However, it also detects false accident in several regions which are actually normal which leads to high false alarms rate. Although, with the help of the complementary information from the trajectories of the moving vehicle these alarms are further refined.

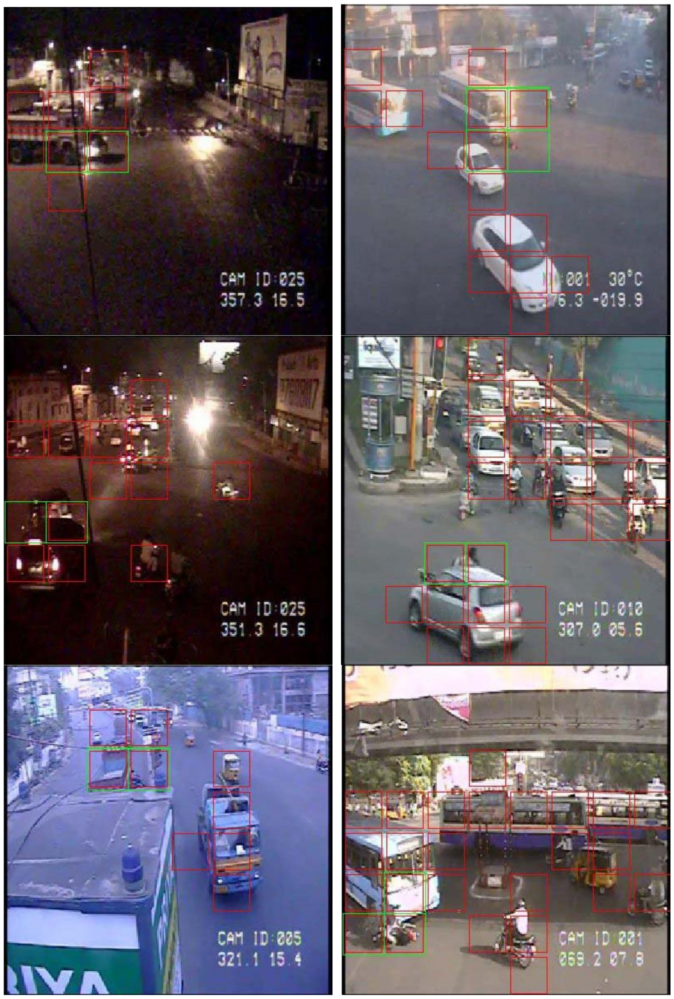


Fig. 11. Accident detected using proposed approach for different videos. The red region is the predicted accident regions using a single score either by appearance, motion, or intersection of tracks while the green box shows the region decided using final score. [Best viewed in color].

Finally, the proposed method can localize the accident events as we are using STVVs instead of entire frame or full video clip. Also, on the collected video dataset of real accidents which contains accidents in various lighting conditions as day, high sun and night it is giving on average 0.775 detection rate at equal error rate (EER) of 0.225.

C. Comparision With the Existing Methods

Instead of a highly desirable task, there is a limited work done in this domain due to unavailability of the public benchmark dataset. Since the existing methods use a small private collection of datasets and do not make them public so comparing them may not be fair at this stage. But still, we listed the performance achieved by the existing methods on individual datasets. ARRS [18] achieve 63% detection rate and 6% false alarms. RTADS [17] achieve 92% detection rate and 0.77% false alarms. The method of Sadek *et al.* [15] shows a recognition rate 99.6% with false alarm rate at 5.2%. Yun *et al.* [10] achieves 0.8950 AUC. However, all the above methods can easily lead to over-fitting for limited samples and do not guarantee the same performance for new scenarios. While, our method is generalized, robust to the over-fitting, and tested on the real traffic with various challenges in the videos. The dataset is made public for the research community for further comparison.

V. Conclusion

The incorporation of convolutional auto-encoder for deep feature representation in proposed framework for accident scene recognition outperforms the existing hand-crafted features based approaches. The method is further strengthened using complementary appearance and motion information together. The dual measures of the outlier scores and reconstruction error for detection of the accidents using complimentary modalities based on appearance, motion, and joint representation increase detection rate of the accidents. The incorporation of the collision of the intersection points of a vehicle's track reduce the false alarm rate, and thus enhances the reliability of the overall system. Since we are using STVVs instead of entire frame or full video clip, it not only detects the accident but also able to localize the accident events. The proposed method is able to detect on average 77.5% accidents correctly with 22.5% false alarms on real accidents videos captured under various lighting conditions. The experimental results are encouraging and show the efficacy of the proposed approach. However, challenges such as low visibility at night, occlusions, and large variations in the normal traffic pattern still pose significant challenges which need to be addressed in future

References

[1] V. Kostakos, T. Ojala, and T. Juntunen, "Traffic in the smart city: Exploring city-wide sensing for traffic control center augmentation," *IEEE Internet Comput.*, vol. 17, no. 6, pp. 22–29, Nov. 2013.

[2] I. Celino and S. Kotoulas, "Smart cities [Guest editors' introduction]," *IEEE Internet Comput.*, vol. 17, no. 6, pp. 8–11, Nov./Dec. 2013, doi: 10.1109/MIC.2013.117.

[3] *European Initiative on Smart Cities, 2010–2020*. Accessed: May 15, 2018. [Online]. Available: https://setis.ec.europa.eu/set-plan-implementation/technology-roadmaps/european-initiative-smart-cities

[4] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy, "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 125–151, Mar. 2015.

[5] A. Dopfer and C.-C. Wang, "What can we learn from accident videos?" in *Proc. Int. Autom. Control Conf. (CACS)*, Nantou, Taiwan, Dec. 2013, pp. 68–73.

[6] C. Regazzoni, A. Cavallaro, Y. Wu, J. Konrad, and A. Hampapur, "Video analytics for surveillance: Theory and practice," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 16–17, Sep. 2010.

[7] G. Yuan, X. Zhang, Q. Yao, and K. Wang, "Hierarchical and modular surveillance systems in ITS," *IEEE Intell. Syst.*, vol. 26, no. 5, pp. 10–15, Sep./Oct. 2011.

[8] S. Xia, J. Xiong, Y. Liu, and G. Li, "Vision-based traffic accident detection using matrix approximation," in *Proc. Asian Control Conf. (ASCC)*, Kota Kinabalu, Malaysia, May/Jun. 2015, pp. 1–5.

[9] J. Ren, Y. Chen, L. Xin, J. Shi, B. Li, and Y. Liu, "Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 428–437, Aug. 2016.

[10] K. Yun, H. Jeong, K. M. Yi, S. W. Kim, and J. Y. Choi, "Motion interaction field for accident detection in traffic surveillance video," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Stockholm, Sweden, Aug. 2014, pp. 3062–3067.

[11] H. Tan, J. Zhang, and J. Feng, "Vehicle speed measurement for accident scene investigation," in *Proc. IEEE 7th Int. Conf. E-Bus. Eng.*, Shanghai, China, Nov. 2010, pp. 389–392.

[12] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

[13] Z. Hui, X. Yaohua, M. Lu, and F. Jiansheng, "Vision-based real-time traffic accident detection," in *Proc. 11th World Congr. Intell. Control Autom. (WCICA)*, Shenyang, China, Jun./Jul. 2014, pp. 1035–1038.

[14] I. J. Lee, "An accident detection system on highway using vehicle tracking trace," in *Proc. Int. Conf. ITC Converg. (ICTC)*, Seoul, South Korea, Sep. 2011, pp. 716–721.

[15] S. Sadek, A. Al-hamadiy, B. Michaelisy, and U. Sayed, "Real-time automatic traffic accident recognition using HFG," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, Istanbul, Turkey, Aug. 2010, pp. 3348–3351.

[16] Ö. Aköz and M. E. Karsligil, "Video-based traffic accident analysis at intersections using partial vehicle trajectories," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 4693–4696.

[17] J.-W. Hwang, Y.-S. Lee, and S.-B. Cho, "Hierarchical probabilistic network-based system for traffic accident detection at intersections," in *Proc. 7th Int. Conf. Ubiquitous Intell. Comput. 7th Int. Conf. Autonomic Trusted Comput. (UIC/ATC)*, Shaanxi, China, Oct. 2010, pp. 211–216.

[18] Y. K. Ki and D. Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 188–194, Jun. 2007.

[19] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 935–942.

[20] W. Sultani and J. Y. Choi, "Abnormal traffic detection using intelligent driver model," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, Istanbul, Turkey, Aug. 2010, pp. 324–327.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1–9.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–14.

[23] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, 2015, pp. 1495–1503.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2016, pp. 3431–3440.

[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[26] N. Li, H. Guo, D. Xu, and X. Wu, "Multi-scale analysis of contextual information within spatio-temporal video volumes for anomaly detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 2363–2367.

[27] J. Jin, A. Dundar, J. Bates, C. Farabet, and E. Culurciello, "Tracking with deep neural networks," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2013, pp. 1–5.

[28] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 597–606.

[29] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.

[30] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4293–4302.

[31] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Swansea, U.K., 2015, pp. 8.1–8.12.

[32] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *J. Mach. Learn. Res.*, vol. 15, pp. 2949–2980, Oct. 2014.

[33] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 1–8.

[34] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann, "Double fusion for multimedia event detection," in *Proc. Int. Conf. Adv. Multimedia Modeling*, Klagenfurt, Austria, 2012, pp. 173–185.

[35] J. Kwon and K. M. Lee, "Wang-Landau Monte Carlo-based tracking methods for abrupt motions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 1011–1024, Apr. 2013.

[36] M. K. Lim, C. S. Chan, D. Monekosso, and P. Remagnino, "Refined particle swarm intelligence method for abrupt motion tracking," *Inf. Sci.*, vol. 283, pp. 267–287, Nov. 2014.

[37] Y. Su, Q. Zhao, L. Zhao, and D. Gu, "Abrupt motion tracking using a visual saliency embedded particle filter," *Pattern Recognit.*, vol. 47, no. 5, pp. 1826–1834, 2014.

[38] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 367–386, Mar. 2015.

[39] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[40] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3169–3176.

[41] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 3551–3558.

Dinesh Singh received the B.Tech. degree from R. D. Engineering College Ghaziabad, Ghaziabad, India, in 2010, and the M.Tech. degree in computer engineering from National Institute of Technology Surat, India, in 2013. He is currently pursuing the Ph.D. degree in computer science and engineering with IIT Hyderabad, India. He joined the Department of Computer Science and Engineering, Parul Institute of Engineering and Technology, Vadodara, India, as an Assistant Professor, from 2013 to 2014. His research interests include machine learning, big data analytics, visual computing, and cloud computing.

Chalavadi Krishna Mohan received the Ph.D. degree in computer science and engineering from IIT Madras, India, in 2007.

He received the Bachelor of Science Education (B.Sc.Ed.) degree from Regional Institute of Education, Mysore, India, in 1988, the M.C.A. degree from S. J. College of Engineering, Mysore, India in 1991, and the M.Tech. degree in system analysis and computer applications from National Institute of Technology Karnataka, Surathkal, India, in 2000.

He is currently an Associate Professor with the Department of Computer Science and Engineering, IIT Hyderabad, India. His research interests include video content analysis, pattern recognition, and neural networks.