Abnormal Event Detection on BMTT-PETS 2017 Surveillance Challenge

Kothapalli Vignesh
Department of EEE
IIT Guwahati
Guwahati, India
k.vignesh@iitg.ernet.in

Gaurav Yadav
Department of EEE
IIT Guwahati
Guwahati, India
g.yadav@iitg.ernet.in

Amit Sethi
Department of EEE
IIT Guwahati
Guwahati, India
amitsethi@iitg.ernet.in

Abstract

In this paper, we have proposed a method to detect abnormal events for human group activities. Our main contribution is to develop a strategy that learns with very few videos by isolating the action and by using supervised learning. First, we subtract the background of each frame by modeling each pixel as a mixture of Gaussians(MoG) to concatenate the higher order learning only on the foreground. Next, features are extracted from each frame using a convolutional neural network (CNN) that is trained to classify between normal and abnormal frames. These feature vectors are fed into long short term memory (LSTM) network to learn the long-term dependencies between frames. The LSTM is also trained to classify abnormal frames, while extracting the temporal features of the frames. Finally, we classify the frames as abnormal or normal depending on the output of a linear SVM, whose input are the features computed by the LSTM.

1. Introduction

Nowadays significant amount of research is being done in the field of automated video surveillance for personnel and asset safety. The need for automation of analysis of surveillance video is increasing day-by-day to reduce the manual workload. Person detection, tracking, activity recognition and event recognition are the areas where researchers are focusing. Working with multiple cameras has its own set of challenges such as tracking a person from one camera to other, group activity recognition etc. Public areas can be monitored with automated systems with less man power. However, detecting an abnormal event from a given video is a challenging problem that is highly contextual.

We propose a method to design a trainable surveillance system to enable situational awareness and determination of potential threats on mobile assets (e.g. vehicles) in transit. BMTT-PETS 2017 Surveillance Challenge is aimed at identifying various abnormal activities which occur in real world scenarios. PETS 2016 dataset [15], also known as ARENA dataset, is used in this paper. The dataset scenarios were recorded from multiple cameras mounted on a vehicle (truck) and involve multiple people demonstrating the event.

There are various challenges in activity recognition in naturally captured videos. Just a one-minute-long video of frame size 640×480 and captured at 30 frames per second can have about half a billion pixels. For the same action class, color and intensities of these pixels can change due to variations in environments, viewpoints, noise, and actor movements. Variations in environments are caused by moving background, occlusion, lighting, and co-occurring of actions of interest with confounding secondary activities. Videos of the same action class taken from different viewpoints have high intra-class variation. For example, a video of walking a dog on grassy background has more pixels in common with that of sport being played on a grassy field than with that of walking a different dog on an urban street. Further, many action classes appear similar, such as walking, jogging, and running, which differ mostly in speed and stride length. Lighting and sensor differences also contribute to variations in pixels of videos of the same action. Common video transformations such as compression and scaling while storing or uploading the video also add to variations in videos of the same action. Additionally, actions of the same class are seldom performed identically in space and time even by the same actor.

In this paper, we are focusing on one of the atomic activities given in the ARENA dataset. The dataset covers a variety of tasks involving low-level video analysis (detection, tracking), mid-level analysis (simple event detection) and high-level analysis (complex threat event detection).

The remainder of this paper is organized as follows: Section 2 describes the state-of-the-art methods, Section 3 describes the proposed method followed by experiment and result in Section 4 and conclusion in Section 5.

2. Related work

There are different methods for activity recognition, which can be divided into three categories: motion-based, shape-based and deep learning-based.

Motion-based approaches are simpler and easier to compute compared to shape-based approaches. These can be further divided into trajectory-based, spatio-temporal volume-based, and region-based approaches [1]. Trajectory based methods extract and cluster trajectories of persons in videos [11]. Tracking itself is a challenging task for noisy videos. To alleviate this problem, spatio-temporal volume-based methods combined with bag-of-words were developed due to their computational efficiency based on reducing the data to a sparse set of salient locations.

Interest points have also played a significant role in development of motion-based video analysis techniques. Wang et al. proposed an interest point-based feature extraction method by forming a group of trajectories of interest points and computing histogram of oriented-gradient (HOG), histogram of flow (HOF), and motion-boundary histogram (MBH) [25]. This approach gives very dense set of interest points. Tracking each point in temporal dimension is computationally expensive. A variation of this theme is to compute SIFT interest points on a frame using hierarchical spatial information, and track them in time [21]. In [30], authors proposed an interest point detector based on differential motion. Each interest point was tracked using KLT tracker [10, 19] and trajectories were extracted, which is computationally expensive. Based on the shape of the trajectories a feature was extracted which was used for classification using SVM classifier. In [18, 29] an interest point detector and SVM were used for action recognition. Dollar et al. [5] proposed an efficient approach for detecting spatio-temporal interest points using a temporal Gabor filter and a spatial Gaussian filter, but it gives too sparse a set of points, which affects the recognition accuracy. In approaches based on space-time volume, generally features are extracted from a volume around spatio-temporal interest points and a bag-of-words model [12] is used to extract the feature. In region-based methods, feature extraction is done for a region of interest. Weinland et al. [27] proposed motion history volumes for feature extraction, and used Mahanolobis distance for classification. Davis proposed a method for computing dense motion flow from motion history images [4]. These techniques depend on the number of feature detected being neither too sparse, nor too dense, which can be subjective.

Multiple pedestrain tracking by using overlapping cameras have been discussed in [24, 7]. In [17], several major challenges in distributed video processing were discussed. They have mentioned various issues such as robust and computationally efficient inference and opportunistic and parsimonious sensing. Largescale video networks start to play an important rule for video surveillance, object recognition, abnormal event detection, and people tracking in crowd environments. Cui et.al. [3], have proposed a method that tracked the local spatiotemporal interest points, and the abnormal activity was indicated by uncommon energy-velocity of the feature points. In [26], based on histogram of optical flow a spatio-temporal descriptor was computed. Bag-of-words model- [12] was used to extract the final video representation which was used for classification. In [25], dense points were sampled from each frame and were tracked based on displacement information from an optical flow field. Then descriptors were extracted based on histogram of flow, histogram of gradient and motion boundary histogram for classification purpose.

Shape-based approaches extract the silhouettes or skeletons to recognize actions. These approaches aim to segment foreground and background to extract the contours of human actions. For example, a viewpoint-independent silhouette-based human action recognition was proposed by Orrite et al. [14]. Each action template was projected onto a new subspace by means of the Kohonen self-organizing feature map and action recognition was done using a maximum likelihood (ML) classifier. The limitation of this method is that it was based on fixed camera settings. Hence, it was easy to extract silhouettes compare to moving background. In [2], the authors proposed a method based on contour points of the silhouettes to represent different poses. Contour points were obtained by applying an algorithm proposed by Suzuki [22]. The center of mass of the silhouettes contour points was calculated with respect to the number of points. The distance signal was generated by determining the Euclidean distance between each contour point and the center of mass. Pose learning was done using k-means clustering and Euclidean distance. It gives real-time performance. However, they have not taken view-invariance into account. Human silhouettes were extracted using background subtraction which assumes that the camera is fixed. Wu et al. [28] exploited the correlation between poses and bag-of-words model for feature extraction. To encode temporal structure information, correlation histogram of human poses in an action sequence was introduced. In all shape-based methods, contour points need to be detected to form the skeleton or to capture the pose. Due to fractured silhouettes and overlapping body parts exact shape extraction is difficult. When there are other moving objects in a scene, extracting the silhouette of an object of interest can be particularly challenging.

Deep learning is also gaining popularity for action recognition. Some approaches based on convolutional neural networks (CNNs) have shown a lot of improvement over state-of-the-art methods due to the success of CNNs for image classification. In [16], the author has proposed an image-based CNN feature for action classification [9] over handcrafted features. CNN requires a large number of samples

for training. As videos are very high dimensional data, training a deep CNN architecture can take a long time. Combination of CNN and SVM was also used in [23, 13], where CNN was used for feature extraction and SVM for recognition. On the other hand, handcrafted features may not give high accuracy but classifiers trained on them can take a lot less time to train compared to a CNN. Our method gives accuracy close to CNN while taking much less time to train. Our approach is also based on CNN.

3. Proposed method

We have focused on a single atomic level activity detection given in BMTT-PETS surveillance dataset which is "person falling or pushed to the ground". In this section we discuss the proposed method as shown in Fig. 1. Due to the small number of videos available for this activity we had to make several changes to the usual ways of applying deep learning. We have filtered uninformative parts of the videos, followed by supervised higher-order feature extraction. First we subtract the background to remove unwanted static visual features. Then we use a Convolutional Neural Network for feature extraction followed by an LSTM network for sequence learning.The output of the last fully connected layer in CNN is passed as an input to the LSTM network. Finally, we use a linear SVM to get the classification scores. To further improve the discrimination between normal and abnormal frames that the entire network has already predicted, we have performed temporal averaging on the final predicted scores. Our method demonstrates how non-deep learning-based methods such as background subtraction and linear SVMs can be combined with deep learning frameworks such as CNNs and LSTMs to build machine learning systems with less data.

3.1. Background subtraction

Mixtures of Gaussians are used for background subtraction as proposed by [8]. Each pixel in the scene is modeled by a mixture of K Gaussian distributions. The probability that a certain pixel has a value of X_N at time N can be written as:

$$p(x_N) = \sum_{j=1}^{K} w_j \eta(x_N; \theta_j) \tag{1}$$

where W_k is the weight parameter of the k^{th} Gaussian component. $\eta(x; \theta_k) = \eta(x; \mu_k, \Sigma_k)$ is the normal distribution of the k^{th} component represented by:

$$\eta(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} \tag{2}$$

where μ_k is the mean and $\Sigma_k = \sigma_k^2 I$ is the covariance of the k^{th} component.

The first B distributions are used as mixtures of the background model of the scene, where B is estimated as:

$$B = \arg\min_b \left(\sum_{j=1}^{b} w_j > T \right) \tag{3}$$

The threshold T is the minimum fraction of the background model. In other words, it is the minimum prior probability that the background is in the scene. Background subtraction is performed by marking a foreground pixel as any pixel that is more than 2.5 standard deviations away from any of the B distributions. The first Gaussian component that matches the test value will be updated by the following update equations:

In Fig. 4 images are shown after background subtraction. We can clearly see that the moving object has been separated from the background, which removes unwanted information.

3.2. Spatial feature extraction

Features from the raw input frames are extracted using a deep CNN with the same architecture as VGG-16 [20] as shown in Fig. 2. This network is employed to extract spatial features as well as for high accuracy image recognition, which is crucial when the frames have distinguishable abnormalities or objects. The network contains 16 trainable (convolutional and fully connected) layers along with several static pooling and dropout Layers. In [20], the authors have shown that the depth of the network plays an important role during its performance. Unnecessary addition of extra layers may not significantly improve the performance while increasing the computational complexity.

3.3. Temporal feature extraction

To further improve the model and recognition accuracy, we make use of temporal relationship between the frames by passing the output of last fully connected layer as input to an LSTM network as shown in Fig. 3. An LSTM specializes in learning long-range dependencies while being unaffected by diminishing or exploding gradient issues that affect earlier recurrent neural networks when trained using backpropagation through time. The non-linear activation gates manipulate the amount of information being stored in the memory cells. Our LSTM architecture consists of two layers. The first layer is made up of 1024 hidden units followed by a bi-directional second layer with 512 hidden units. The standard recurrent neural networks have restrictions on the amount of data that is provided as input. i.e. they have limitations on the input data flexibility. In such networks the future input information cannot be reached from the current state. To overcome this problem, we use a bidirectional LSTM where we have a forward LSTM and a backward LSTM running in reverse time with

Figure 1. Block diagram of the proposed method.

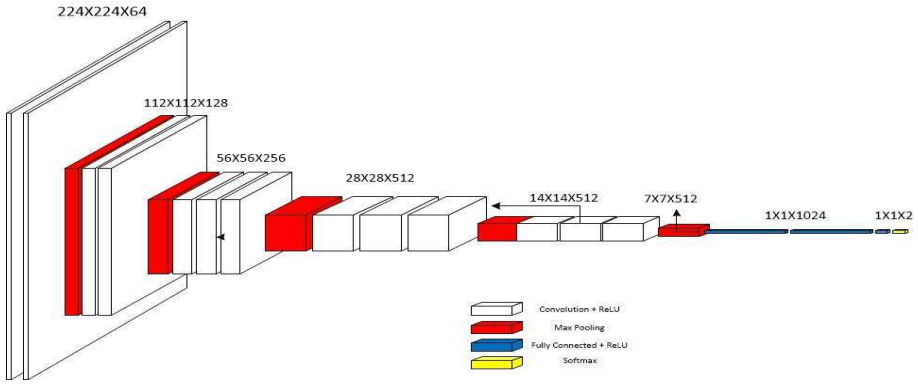


Figure 2. CNN Architecture based on VGG16.

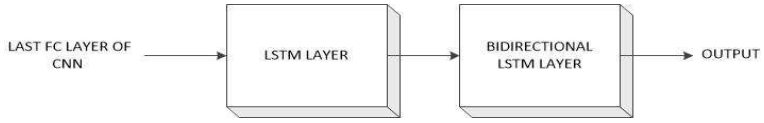


Figure 3. LSTM architecture.

their features concatenated at the output layer, thus allowing us to combine the useful features of the past and the future.

3.4. Classification

For classification of the event frames we tried two methods: average fusion of the networks and a linear SVM classifier. While classifying using average fusion, we have averaged the outputs of CNNs and LSTMs softmax layers and predicted the abnormalities based on the scores. We also tried a linear SVM classifier. SVM has been proven to be a state-of-the-art linear classifier which maximizes the margin between two classes. More importantly, it works well with high-dimensional and low sample size datasets. We have trained the SVM using one-vs-all approach using scikit learn libraries [6].

4. Experiment and Results

In this section, we describe the datasets, experimental setting for training and testing followed by recognition results.

4.1. Datasets

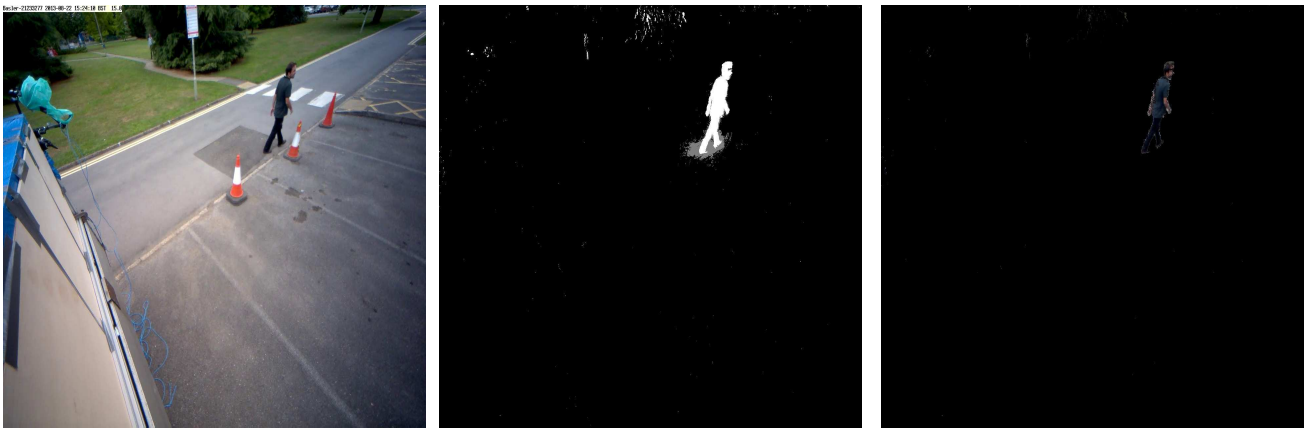
The provided dataset is a multi-sensor dataset, as used for PETS 2014 to PETS 2016, which addresses protection of trucks (the ARENA Dataset). The dataset includes a set of abnormal events such as person falling on ground, person speeding up, person loitering, person suddenly changing directions etc. We focus only on the 'person falling or pushed to the ground' event detection. There are three folders – 11-04, 11-03 and 08-02 which depict the different ways in which a person can fall on the ground. Each folder has frame sequences of videos from four different cameras mounted on the truck. We have classified the frames into two classes – 'normal' and 'abnormal'(where the person of interest is falling) and labeled them. The task was to predict the starting and ending frames of the abnormal activity sequence that is taking place. For cross validation or testing, the data was split at the folder level because each folder contained video frames from a different scene (e.g. parking location, background of the truck etc).

Table 1. Labeling of the Start and End frames for each folder

Folder	Start frame	End frame
11-04 TRK-RGB1	1377185170222	1377185175622
11-03 TRK-RGB1	1377185040756	1377185045090
08-02 TRK-RGB2	1377181598164	1377181608514

4.2. Results and Discussion

We have experimented with different combinations of CNN, LSTM, and SVM. Each combination is explained as follows:



(a) (b) (c)

(d) (e) (f)

Figure 4. Example frames for Background subtraction, (a)-(d) Original frame (b)-(e) Binary image after background subtraction, (c)-(f) Background subtraction overlapped with original.

Table 2. Performance comparison on the given data-sets.

Method	11-04	11-03	08-02
CNN	83.3%	80%	80%
CNN+SVM	83.8%	80.7%	80.1%
CNN+LSTM	84.2%	81.3%	80.2%
CNN+LSTM+SVM	85.3%	83.4%	82.9%
CNN+LSTM+SVM+TA	85.3%	96.4%	94.9%

Table 3. Start and end frame for CNN-LSTM-SVM-TA.

Folder	Start frame	End frame
11-04 TRK-RGB1	1377185162089	1377185163755
11-03 TRK-RGB1	1377185040889	1377185045356
08-02 TRK-RGB2	1377181598664	1377181606314

4.2.1 CNN

We have used the CNN architecture as shown in Fig. 2 and tested on folders 11-04 (TRK-RGB1), 11-03 (TRK-RGB1), 08-02 (TRK-RGB2) consisting of 729, 329, 1056 frames respectively, taking into consideration only the cameras which capture the abnormal activity that is occurring. The VGG-16 CNN architecture is first trained from scratch on 11-04 (TRK-RGB1 and TRK-RGB2) and 11-03 (TRK-RGB1 and TRK-RGB2) data and then tested on 08-02 (TRK-RGB2). We repeat the same process by initializing the model with new weights but by training on 08-02(TRK-RGB1 and TRK-RGB2), 11-04 (TRK-RGB1 and TRK-RGB2) from scratch and testing the model on 11-03 (TRK-RGB1). Similarly, we train on 11-03(TRK-RGB1 and TRK-RGB2), 08-02(TRK-RGB1 and TRK-RGB2) and test on 11-04 (TRK-RGB1). The results for all the cases are mentioned in the Table 2.

Discussion: We have not used pre-trained weights for VGG-16 convnet because, the pre-trained model has already adapted itself to the huge number of classes (1000) and it would be difficult for the network to learn the weights when we are performing a binary classification with a considerably small dataset. Thus, training the network from

scratch gave better results compared to those using the pre-trained model.

4.2.2 CNN+SVM

In Section 4.2.1 we have used the softmax layer as the output layer of CNN for all training and testing cases. In this section we explore the benefits of adding a support vector machine for classification of the features determined using CNN. The methodology of training and testing remains the same, as mentioned in Section 4.2.1.

Discussion: Since support vector machines are robust to a decrease in the number of training samples for classification, incorporating it as the output layer improved results over softmax as shown in Table 2.

4.2.3 CNN+LSTM

We have incorporated LSTM network on top of our CNN architecture to model the temporal information in the visual channel. LSTMs are one of the most widely used RNN models that learn long-range dependencies of the sequential data by incorporating memory cells. These cells are protected by non-linear gates (with $tanh$ or sigmoidal activation functions) which makes LSTMs immune to vanishing and exploding gradients, unlike the traditional RNNs. These gates control the amount of information that needs to be stored in memory, forgotten and passed on to the next hidden unit or to the output layer. Therefore, after adding LSTM and using softmax as a classifier, we can see the improvement over CNN and CNN-SVM.

Discussion: In the second layer of LSTM network we have used the bi-directional LSTM hidden units instead of the usual hidden units. The reason being: bi-directional LSTM layer acts as a combination of both a forward LSTM and a backward LSTM which runs reverse in time and the features of both are merged to get the output. Through this method, we predict the output based on the information from the past and the future. Therefore, recognition accuracy further increases compared to previous combinations as shown in Table 2.

4.2.4 CNN+LSTM+SVM

Since we are working on a very small dataset compared to other video classification problems consisting of hours of videos as the training data, the usage of SVM's proves to be beneficial. We incorporate a final SVM layer on top of the CNN+LSTM architecture so as to learn the final decision boundary between the normal and abnormal sequences/frames. This gives the best results compare to CNN,CNN-SVM, and CNN-LSTM as shown in Table 2.

Discussion: We now incorporate the best performing networks on top of each other for final classification of the frames. We observe that the proposed classification method performs better than the average fusion method, where the predicted probability outputs of CNN and LSTM softmax layers are merged and averaged to predict the class.

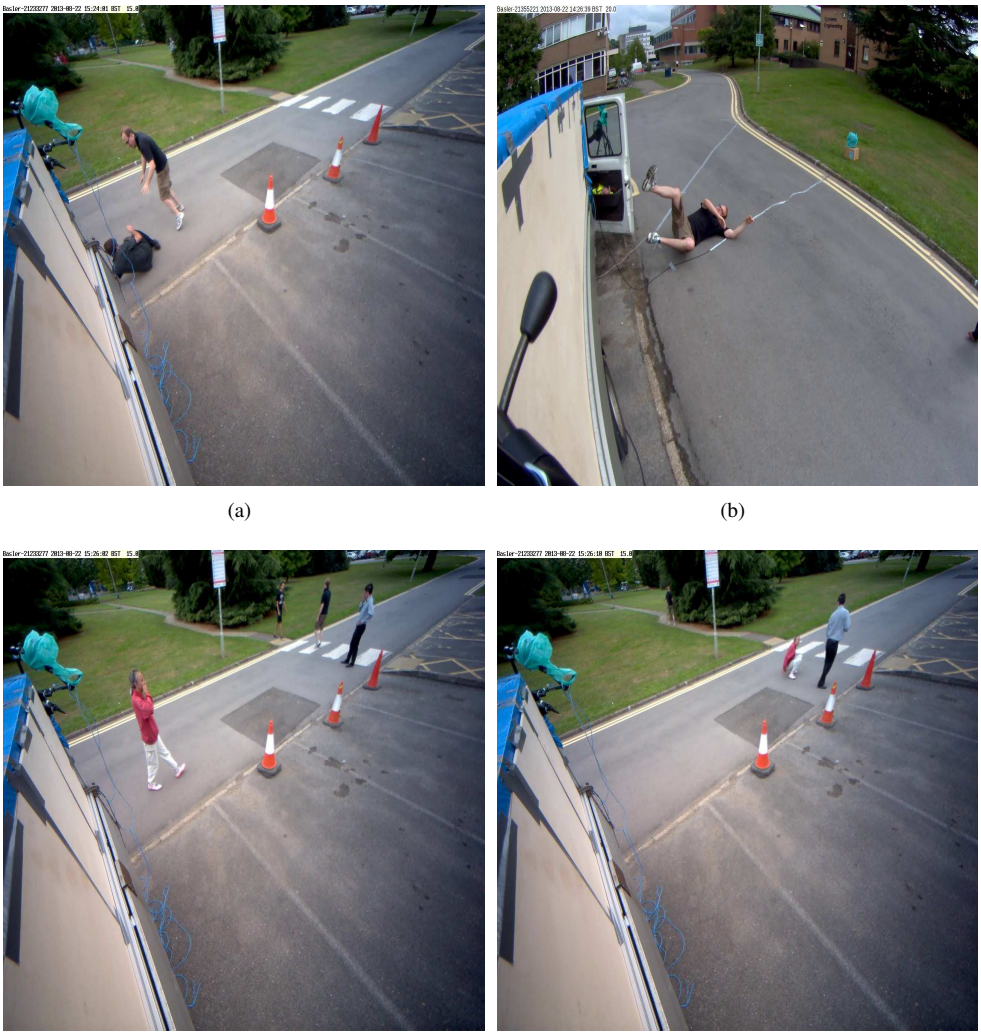
4.2.5 CNN+LSTM+SVM+TA

To remove the discontinuity in the prediction of frames we have used temporal averaging after the CNN+LSTM+SVM model. For temporal averaging we choose a particular length of frames and based on the number of majority predictions, we change the predicted label of the entire length of frames to the majority label.

4.2.6 Results

For training VGG-16 we have used the same setting as author has suggested in [20], but with few changes. The input size is $224 \times 224 \times 3$ and dropout regularization for the two fully-connected layers was 0.5. A change in number of neurons in the last two Fully Connected Layers was made and was set to 1024 instead of 4096. The learning rate was initially set to 10^{-4}, then decreased by a factor of 10. We fine-tuned on any two folders using the normal vs. abnormal frame label, and tested on third one for cross-validation at the folder level. Hence, we get three accuracy number for three folders as shown in the Table 2. From Table 2 we can see that adding LSTM to CNN improves the results and further adding SVM again improves the results. The starting and ending frames are also mentioned in the Table 3. 11-03 and 08-02 have shown correct predictions of frames. But in case of 11-04, it fails to identify the correct frames because it is different form the other two datasets. The model has not seen such type of activity that far away from the camera and for a very short duration, when compared to other two folders. As shown in Fig. 5, the abnormal activity is happening in the vicinity of the vehicle in case of 11-03 and 08-02 unlike 11-04. Therefore, the network is unable to classify correctly in case of 11-04 frames and it classifies those frames as abnormal where the person is walking near the vehicle.

To further improve the classification accuracy and to predict the start and end frames as close as possible to the ground truth, we have performed temporal averaging on the predictions of the above mentioned network (CNN+LSTM+SVM+TA). We observe that the accuracies on 11-03 and 08-02 jumps to 96 and 95 percent respectively, and the predicted start and end frames are very close to the ground truth. This method is not applicable to 11-04 because the initial predictions doesn't cover the abnormal event.



(a) (b)

(c) (d)

Figure 5. Example frames from datasets (a) 11-03, (b) 08-02, (c) 11-04 normal frame and (d) 11-04 abnormal frame.

5. Conclusion

Among all the combinations, CNN-LSTM-SVM-TA model has shown the best performance. CNNs perform best when a large number of training samples are provided for training, whereas for less number of samples it can be used for feature extraction instead of classification. These features mostly contain spatial dependencies among frames but not the temporal ones. Therefore, to overcome these issues, we combined CNN features with an LSTM model and an SVM classifier. LSTM learns temporal relationships between frames and the SVM classifies the frames independent of the number of redundant samples. SVM can thus be used instead of softmax for classification. We have evaluated these models for one type of activity, but it can also be further extended to other activities.

References

[1] C. Cedras and M. Shah. Motion-based recognition a survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[2] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta. Silhouette-based human action recognition using sequences of key poses. *Pattern Recognition Letters*, 34(15):1799–1807, 2013.

[3] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas. Abnormal detection using interaction energy potentials. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3161–3167. IEEE, 2011.

[4] J. W. Davis. Hierarchical motion history images for recognizing human motion. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 39–46. IEEE, 2001.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In

2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pages 65–72. IEEE, 2005.

[6] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training support vector machines. *Journal of machine learning research*, 6(Dec):1889–1918, 2005.

[7] W. Jiuqing and L. Achuan. Multiple people tracking using camera networks with overlapping views. *International Journal of Distributed Sensor Networks*, 2015.

[8] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. Springer, 2002.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[11] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 514–521. IEEE, 2009.

[12] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3):299–318, 2008.

[13] X.-X. Niu and C. Y. Suen. A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

[14] C. Orrite, F. Martínez, E. Herrero, H. Ragheb, and S. Velastin. Independent viewpoint silhouette-based human action modelling and recognition. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis-MLVMA'08*, 2008.

[15] L. Patino, T. Cane, A. Vallee, and J. Ferryman. Pets 2016: Dataset and challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2016.

[16] M. Ravanbakhsh, H. Mousavi, M. Rastegari, V. Murino, and L. S. Davis. Action recognition with image based cnn features. *arXiv preprint arXiv:1512.03980*, 2015.

[17] A. C. Sankaranarayanan, R. Chellappa, and R. G. Baraniuk. Distributed sensing and processing for multi-camera networks. In *Distributed Video Sensor Networks*, pages 85–101. Springer, 2011.

[18] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[19] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2004–2011. IEEE, 2009.

[22] S. Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[23] Q.-Q. Tao, S. Zhan, X.-H. Li, and T. Kurihara. Robust face detection using local cnn and svm based on kernel combination. *Neurocomputing*, 211:98–105, 2016.

[24] J. Wan and L. Liu. Distributed bayesian inference for consistent labeling of tracked objects in nonoverlapping camera networks. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[25] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[26] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, pages 124–1. BMVA Press, 2009.

[27] D. Weinland, R. Ronfard, and E. Boyer. Motion history volumes for free viewpoint action recognition. In *Workshop on modeling People and Human Interaction (PHI'05)*, 2005.

[28] D. Wu and L. Shao. Silhouette analysis-based action recognition via exploiting human poses. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):236–243, 2013.

[29] G. K. Yadav and A. Sethi. A flow-based interest point detector for action recognition in videos. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, page 41. ACM, 2014.

[30] G. K. Yadav, P. Shukla, and A. Sethfi. Action recognition using interest points capturing differential motion information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1881–1885. IEEE, 2016.