IJCTA, 9(21), 2016, pp. 43-49
© International Science Press

Modified Dense Trajectory for Real Time Action Recognition

Vikas Tripathi* Piyush Bhatt Sharat Agarwal*** Monika Semwal** ****

Abstract : This paper introduces an efficient algorithm based on dense point extraction using dense trajectory for real time action recognition. This proposed approach focuses on extraction of dense points from each frame and then tracking them based on the displacement information from a dense optical flow fields. Dense trajectory has been a success due to its efficiency in abnormal description in a video sequence. Our proposed approach reduces the time complexity of the existing algorithm without affecting its efficiency.

Keywords : Action Recognition, Real time, dense trajectory, video surveillance, centroid, UT Interaction and bag of features

1. INTRODUCTION

Abnormal event detection is a very dominant field in research as it is important for public safety and security. As data is increasing at a war footing pace it is very difficult for someone to handle it at the personal level so there is a dire need of an automated system that helps in detecting abnormal events from a video sequence. Previously, a robust method of event classification on the basis of dense points was introduced by Wang et al.[7] that used optical flow to detect motion in the video sequence and thus classifying it. Using approach of classifying events, an improved version of dense trajectory was proposed by Wang et al.[2]. They showed how to enhance performance significantly by removing trajectories of background. For optical flow warping with a robustly estimated homography can be used to increase accuracy of system. The descriptors used in these algorithms outperform the other futuristic descriptors. Laptev and Lindeberg [3] introduced interest points based on spatial and temporal information by extending the approach used in Harris detector. Several other methods are also used to extract interest points like Gabor filters [6, 5] or spatiotemporal Hessian matrix [4].

In classification of images sampling of dense point has generated better results over sparse time interest points [8, 13]. Further in video analysis, activity recognition and in event recognition it has been observed that dense sampling can achieve better accuracy due to large amount of motion involve in video analysis [10]. In [10] they have shown that dense sampling when used in spatiotemporal aspect can outperform other benchmark algorithms based on space-time interest points. The local space-time features and descriptors success have shown new way to generalize traditional descriptors from image to video. Several researchers have made changes in traditional descriptors like Skovannner et al.[9] and have introduced SIFT in three dimension. Williams et al. in [11] proposed extended SURF, three dimensional HOG was explained by KlÓse et al. in [14], and local trinary patterns by Yeffet and Wolf [12]. Robustness to motion of camera incorporated in Motion boundary histograms (MBH) [15] hence, it gives excellent results. In this Clustering of Trajectories in a video is done and further for every cluster center an affine transformation matrix is computed. The trajectories thus are depicted by these matrix elements. Sun et al.

* Department of Computer Science and Engineering Uttarakhand Technical University, Dehradun, Uttarakhand, India Graphic Era University, Dehradun, Uttarakhand, India vikastripathi.be@gmail.com

** Govind Ballabh Pant Engineering College, Pauri Garhwal, Uttarakhand, India bpiyushcse@gmail.com

*** Department of Computer Science and Engineering Graphic Era University, Dehradun, Uttarakhand, India sharat29ag@gmail.com

**** Department of Computer Science and Engineering Graphic Era University, Dehradun, Uttarakhand, India somi.semwal@gmail.com.

[20] have used SIFT descriptor between every two consecutive frames and by matching it they have extract trajectories. In some recent methods [7, 21, 23] accuracy have been improved by incorporating the dense trajectories in action recognition.

In this paper, we are improving the dense trajectory method by reducing the time constraint. This approach is not used before though many works have been done on trajectories. Dense trajectories have not been efficiently utilized in realistic environment in the past for action recognition due to heavy dataset generated by it. Sundaram et al. [17] sped up dense trajectories by making computation on a GPU. Brox et al. [19] sectioned objects by bundling dense trajectories. This method is known as clustering. An identical approach is employed in [16] for extraction of object from video. Distinguishing various actions or activities, motion is the most informative and suitable factor. It can be due to the action of interest, but also wrong information can also be generated due to background or the camera movement. This is certain while dealing with realistic actions where the settings are not under our control. Separating action motion from extraneous motion is still an open problem. Ikizler-Cinbis et al. [18] used motion compensation procedure for better video stabilization. In this the camera motion is eliminated to a greater extent.

This paper is organized as follows. In section 2, proposed approach is explained, followed by the results in section 3. And finally section 4 summarizes the paper.

2. PROPOSED ALGORITHM

2.1. Dataset

The trajectory feature that we use in this experiment is conducted on two datasets namely UT Interaction [22] and ATM (Automated Teller Machine) dataset [23]. These Datasets are diverse and capture some very normal and regular activities. As feature trajectory show significant variation in motion, therefore these datasets are chosen to use the algorithm efficiently.

UT Interaction Dataset

This dataset contains videos of continuous executions of 6 classes of human-human interactions: hand-shake, pointing, hugging, pushing, kicking and punching. There are 20 video sequences of 1 minute and each video contains at least one interaction being executed. In UT Interaction we have taken two classes- normal and abnormal class. We classify the frames for these two classes using the proposed approach. Fig.1. (*a*) illustrates the abnormal events in UT Interaction dataset, *i.e*., kicking, boxing and pushing. Fig.1. (b) illustrates normal events in UT Interaction dataset, *i.e*., hugging, pointing and handshaking. The dataset covers the normal actions that occur in day to day life. These actions are done in different conditions and clothes to make it highly diverse for classification. The two classes are made on the basis of the actions that are harmful (abnormal) and those not harmful (normal).

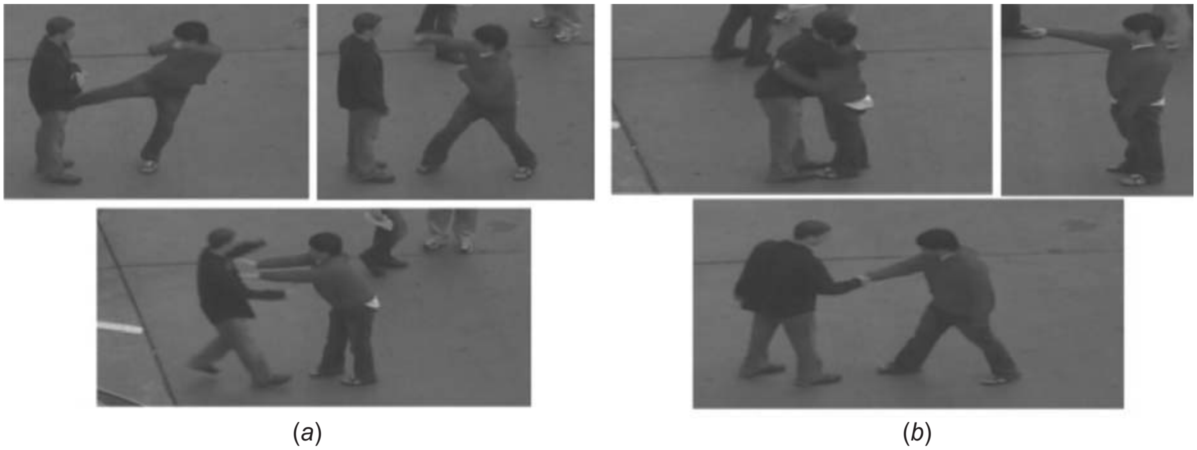
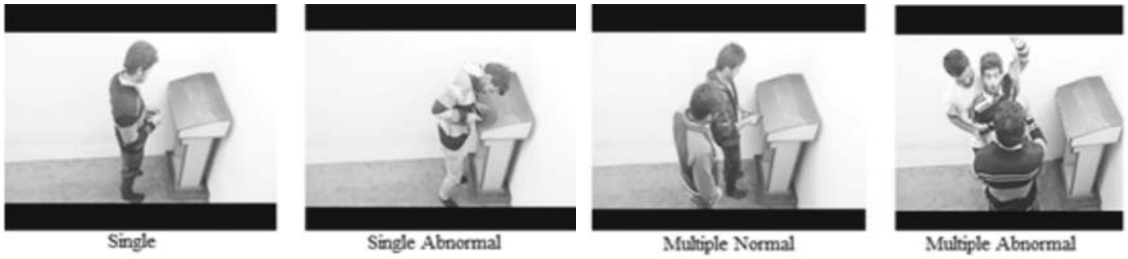


Fig. 1. (*a*) Abnormal event in UT Interaction. Kicking, Boxing and Pushing (Left-Right). (*b*) Normal event in UT Interaction. Hugging, Pointing and Handshaking (Left-Right)

ATM Dataset

We have made our own data set which simulates ATM working with frame size 320x240 and 25fps frame rate. Fig. 2(*a*) and Fig. 2(*b*) shows sample frames of various activities from video sequence of ATM Dataset. The ATM dataset contains four activities single normal, multiple normal, single abnormal, and multiple abnormal over 40 videos. ATM dataset system was trained by using eight videos of each activity and tested against all four classes over eight videos, two videos for each activity.



(*a*) (*b*)

Fig. 2. (*a*) Single normal and abnormal event of ATM dataset. (*b*)Multiple normal and abnormal event from ATM dataset

2.2. Dense Trajectory

Dense sampling has outperformed the sparse interest point classification. In case of action recognition the same has been observed in the evaluation done by Wang et al. [1]. According to them, dense sampling at regular positions in space and time gives better results than the normal space-time interest point detectors.

These dense samples are tracked using optical flow fields to find the trajectories. Increase in the points that are tracked can be done without difficulty because dense flow fields are reckoned before. In addition to this, global smoothness restrictions are enforced amidst the dense points that results in more robust trajectories than tracking and matching points independently. We are incorporating two algorithms into the original dense trajectory algorithm. The two changes that have been done are:

2.2.1. Adding two frames before moving forward to dense trajectory.

As shown in Algorithm1 we added two frames before giving them as an input to the algorithm. This reduced the number of frames that have to be processed by half. In some cases, the dense points might be higher as the added frame depicted more motion but the overall result was efficient. Without compromising on the efficiency that original dense trajectory algorithm gave, we made these two alterations to speed up the process.

Algorithm 1 : Begin read video from the source while(frames! = NULL) add two consecutive frames apply dense trajectory to these modified frames

End

The frames that are added are used for dense point detection. This reduces the dataset by half making it easier to handle.

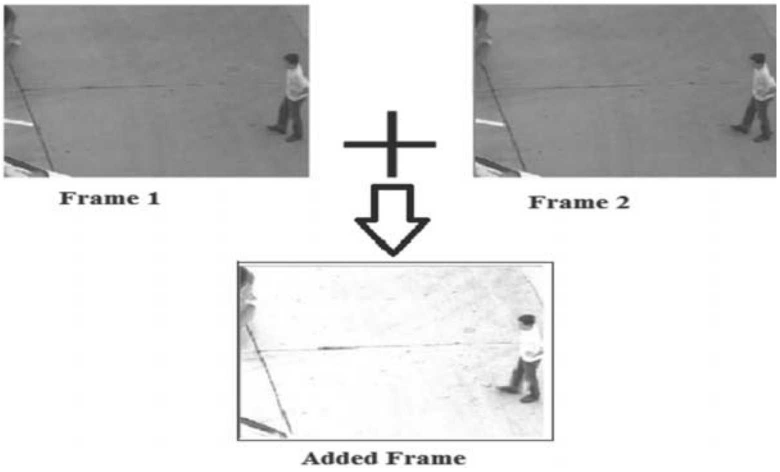


Fig. 3. Illustrating Adding of two frames.

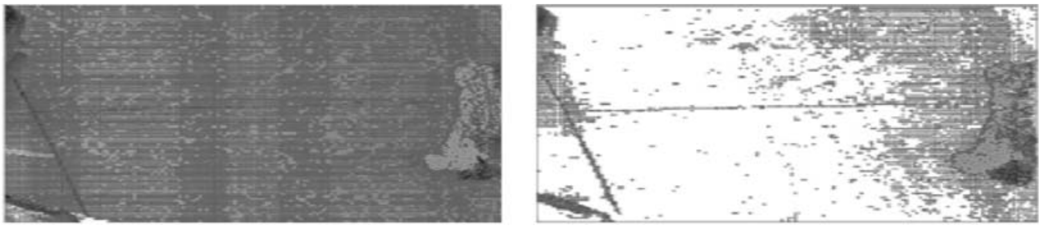


Fig. 4. Illustrating dense points in single and added frames (from left to right) respectively.

The dense points in single frame are comparatively less than in added frame as seen in the figure 4. This is because when we add two frames, there is reduction in multiplicative noise. Multiplicative noise is the unwanted and arbitrary signals that gets multiplied while capturing, transmitting or any other image processing. The reduction in this kind of noise limits the descriptors and makes the dataset less robust.

2.2.2. Skipping dense points / centroids before training and testing

In some frames the dense points detected are very large in number increasing the time and space complexity of algorithm during execution. To avoid this, we skipped some dense points during the process of tracking trajectories as shown in Algorithm 2. The skipping was done only on those frames where the number of dense points was higher than the threshold value defined by us. This helped in expediting the process and saving the time during execution. After testing many possibilities to skip the dense points, the best result was shown when we skipped 2 points after it the number of dense points reached its threshold value of 30. This way we did not lose any information as dense points from all over the frame were chosen.

Algorithm 2:

```
Begin cluster and build the dictionary using selected dataset.
centroid Count = 0
for each frame
read the centroid
ifcentroidCount > 30
skip two centroid
centroid Count = centroid Count + 2
centroid Count = centroid Count + 1
End
```

The output of this algorithm is taken as the dataset for training. Similarly, we skip the centroids using this algorithm while testing.

We tested these two algorithms individually but got the best result when they were both added in the original dense trajectory algorithm. Algorithm 1 was added at the beginning where the frames are read. Before the frames can go to the original dense trajectory algorithm, they were added and then sent to it. Whereas, Algorithm 2 was added after dictionary was made and before the dataset was sent for training and testing. The dataset that was made for training and testing have to go through this algorithm where the centroids dense points were skipped to make the final dataset smaller.

2.3. Bag of Features

Assessing the performance or execution of modified method discussed here is done using standard bag-of-features method. As given in standards firstly construction of codebook is done for every individual descriptor. The descriptors being trajectory, HOG, HOF and MBH. Number of visual word need to be fixed and play important role in accuracy for each descriptor. We fixed 100 visual words per descriptor to limit the complexity. Further K means clustering algorithm is employed to generate centroids. Clustering a subset of 100,000 training features that are randomly selected. Clustering is done using *k*-means.For increasing the accuracy, value of iterations initialized 8 times in k-means and the desired accuracy of 0.001 whichever reaches first.

Euclidean distance is for distance calculation. On the basis of distance, the closest vocabulary word is assigned for the descriptors. Further histograms are calculated of the visual word frequency of occurring and used as video descriptors.

3. EXPERIMENT AND RESULT

This section contains the evaluation of performance of our algorithm and its comparison with the original dense trajectory algorithm. This will clearly give the idea as to how useful our algorithm is compared to the original one. Random forest algorithm is being used in our approach. It is a combination of decision trees. Each of these trees is trained as per feature set provided by picking a set of decision functions to classify the test data. Random forest is robust to noise in training dataset; hence it gives very stable model builder. However, performance is often dependent on dataset. We use Random Forest Classifier with random forest of 100 trees, each constructed while considering 9 random forests with 500 attributes.

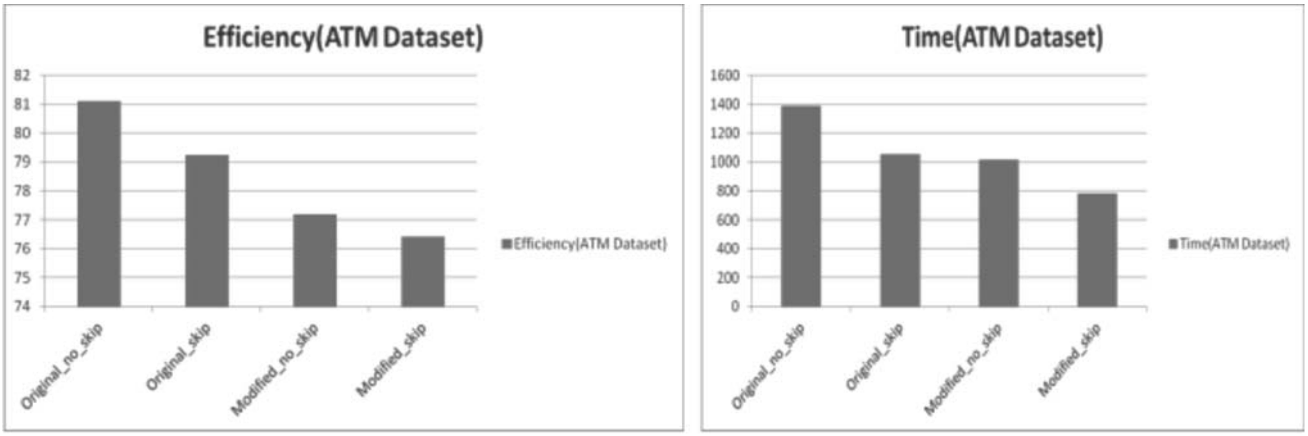
The comparative analysis can be done through the tables given below. The efficiency does not deteriorate much as compared to the improvement in time constraint making the new algorithm faster. Apart from original dense trajectory algorithm and our modified algorithm (addition of two frames and skipping dense points after 30), the other two are the different possibilities that we tested.

In ATM Dataset the original algorithm gave an efficiency of 81.13% with testing time as 1391.4 seconds whereas the modified algorithm gave an efficiency of 76.42% with testing time 785.009 seconds. The decrease in efficiency is 5% while the reduction in testing time is 43.58% which is a good deal as the time is reduced significantly with just a little decrease in efficiency.

Table 1. Comparison of original dense trajectory algorithm and modified algorithm in ATM dataset.

Original/ Modified.	*Skipping Dense Points*	*Efficiency*	*Time taken*
Original	No	81.13%	1391.4 sec
Original	Yes	79.24%	1052.98 sec
Modified	No	77.19%	1017.14 sec
Modified	Yes	76.42%	785.009 sec

Both the algorithms affect the efficiency and time constraint in some way or the other. As we can see in the table 1 the original algorithm (without addition of frames) gives the maximum efficiency among all the other cases but the time taken during testing is also maximum. In the second case, the efficiency and time decreases a bit. The dense points that are skipped is the main reason behind it as more the dense points more is the information extracted from the particular frame. The time taken that is decreased is because now the algorithm has to deal with lesser dense points.



(*a*) (*b*)

Fig. 5. (*a*) Graph representing efficiency of all four cases of ATM dataset. (*b*) Graph representing time taken by all four cases of ATM dataset.

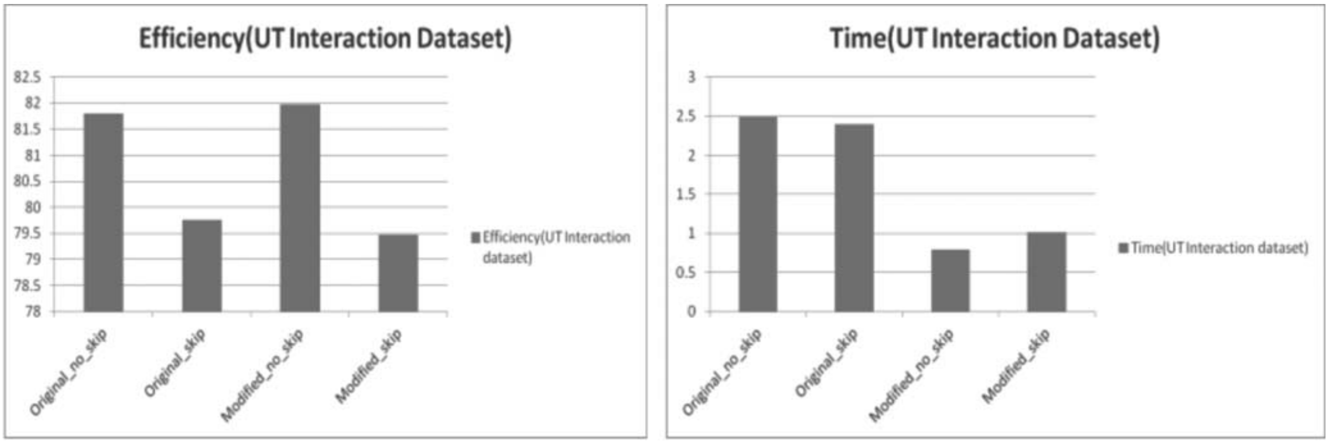
In the third case, we've added two frames and then feed them to the algorithm. As mentioned above, addition of frames reduces multiplicative noise which in turn also reduces the dense points detected. Finally, in the last case we've used both the modifications, so there is maximum difference in efficiency and time between this case and the original algorithm. Graph in figure 5 illustrates it better.

The graphical representation in above two figures shows efficiency and time taken by ATM dataset. Though the modified approach gives lesser efficiency yet the time taken by the modified approach is very less. Therefore, making this modified approach suitable for our real time processing. In UT Interaction dataset, the original algorithm gave an efficiency of 81.8% with testing time as 2.49 seconds whereas the modified algorithm gave an efficiency of 79.48% with testing time 0.79 seconds. The decrease in efficiency is 2.83% while the reduction in testing time is 68.27%. This is much better than the ATM results.

Table 2. Comparisons of original dense trajectory algorithm and modified algorithm in UT interaction dataset.

Original/ Modified	*Skipping the Dense Points*	*Efficiency*	*Time taken*
Original	No	81.8%	2.49 sec
Original	Yes	79.76%	2.39 sec
Modified	No	81.97%	1.01 sec
Modified	Yes	79.48%	0.79 sec

The outcome of three cases in UT interaction dataset is same as that in ATM dataset except for the third case. This case gives the best result even after addition of frames. It is because this dataset is an organized one in which every action is clear. There are no complex activities that can confuse the algorithm into giving a wrong output. As there are no speedy motions happening, the added image does depict motion from one frame to another. As in case of ATM dataset, there is random motion happening and the added frame is saturated with motion.



(*a*) (*b*)

Fig. 6. (*a*) Graph representing efficiency of all four cases of UT Interaction dataset. (*b*) Graph representing time taken by all four cases of UT Interaction dataset.

The graphical representation in the above figures 6(*a*) and 6(*b*) clearly show that time has reduced significantly in modified approach.

4. CONCLUSION

In this paper we have proposed a much efficient method for real time action recognition by improving an already existing algorithm of robust dense trajectory. Our approach has effectively showed much better results in terms of reducing time complexity but very slightly compromising in efficiency which can be negotiated in cases of large database, as this new algorithm is efficient enough to handle larger datasets. We have improved effectiveness in terms of time taken by 63% compromising of 6.8% in efficiency. This kind of portrayal has shown to be potent for action classification, but could also be used in other areas, such as action localization and video retrieval. The dense trajectory descriptors (HOG, HOF and MBH) combine together to provide trajectory shape, appearance, and motion information. There is lot of scope in improving the both the efficiency and time complexity of algorithm in future and much work can be done.

5. REFERENCES

1. H. Wang., M. M.Ullah, A Klaser.,I. Laptev, and C.Schmid, “Evaluation of local spatio-temporal features for action recognition”, In BMVC British Machine Vision Conference, pp. 124-1,2009.
2. H.Wang, ,D. Oneata,J. Verbeek, and C. Schmid, “A robust and efficient video rep-resentation for action recognition”, arXiv preprint arXiv:1504.05524, 2015.
3. I. Laptev, “On space-time interest points”, International Journal of Computer Vision, vol. *64,pp*. 107-123, 2005.
4. G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector”, In *Computer Vision–ECCV,* pp. 650-663,2008.
5. P. Dollár, V. Rabaud, G.Cottrell and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2nd Joint IEEE International Workshop,* pp. 65-72., 2005.
6. M. Bregonzio, S. Gong, and T. Xiang, “Recognising action as clouds of space-time interest points”, In *Computer Vision and Pattern Recognition,CVPR, IEEE Conference,* pp. 1948-1955, 2009.
7. H. Wang., A. Kläser, C. Schmid, and C. L. Liu, “Action recognition by dense trajectories”, In *Computer Vision and Pattern Recognition (CVPR),* pp. 3169-3176.2011.
8. L. Fei-Fei, and P. Perona, “A bayesian hierarchical model for learning natural scene categories”, In *Computer Vision and Pattern Recognition,CVPR, v*ol. 2, pp. 524-531, 2005.
9. P. Scovanner, S. Ali, and M. Shah , “A 3-dimensional SIFT descriptor and its application to action recognition”, In: ACM Conference on Multimedia, 2007.
10. G. Willems, T. Tuytelaars, and L. Gool, “An efficient dense and scaleinvariantspatio-temporal interest point detector”, European Conference on Computer Vision, pp. 650-663, ,2008.
11. L. Yeffet,L.Wolf, “Local trinary patterns for human action recognition”,In: IEEE International Conference on Computer Vision, pp. 492-497,2009.
12. E. Nowak ,F. Jurie, and B. Triggs, “Sampling strategies for bag-of-features image classification”, In *Computer Vision–ECCV,* pp. 490-503, 2006.
13. A. KlÓser, M. Marsza³ek, C. Schmid, “A spatio-temporal descriptor based on 3D-gradients”, In: British Machine Vision Conference, pp. 275-1, 2008.
14. N. Dalal, B.Triggs, C. Schmid, “Human detection using oriented histograms of flow and appearance”, In: European Conference on Computer Vision, pp. 428-441, 2006.
15. Lu, W. C., Wang, Y. C. F., and C.S. Chen, , “Learning dense optical-flow trajectory patterns for video object extraction”, In *Advanced Video and Signal Based Surveillance (AVSS), pp. 315-322*, 2010.
16. N. Sundaram, T. Brox, and K. Keutzer, “Dense point trajectories by GPU-accelerated large displacement optical flow”, In *Computer Vision–ECCV,* pp. 438-451, 2010.
17. N. Ikizler-Cinbis, and S. Sclaroff, “Object, scene and actions: Combining multiple features for human action recognition”, *Computer Vision–ECCV* , pp. 494-507, 2010.
18. T. Brox, and J. Malik , “Object segmentation by long term analysis of point trajectories”, In *Computer Vision–ECCV,* pp. 282-295, 2010.
19. 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,* pp. 2004-2011, 2009.
20. R. Messing, C. Pal, and H. Kautz, “Activity recognition using the velocity histories of tracked keypoints”, In *Computer Vision, IEEE 12th International Conference,* pp. 104-111, 2009.
21. P. Matikainen, M. Hebert, and R. Sukthankar, “Trajectons: Action recognition through the motion analysis of tracked features”, In *Computer Vision Workshops, ICCV Workshops, 12th International Conference,* pp. 514-521. 2009.
22. M. S. Ryoo, and J. K. Aggarwal. “UT-interaction dataset, ICPR contest on semantic description of human activities (SDHA).” *IEEE International Conference on Pattern Recognition Workshops*. Vol. 2. 2010.
23. V. Tripathi, A. Mittal, D. gangodkar, and V. Latta, “Robust abnormal event recognition via motion and shape analysis at ATM installations.” *Journal of Electrical and Computer Engineering*. 2015.