ISSN: 0369-8963

# An Ensemble Learning and Swarm Based Segmentation Framework for Video Event Recognition

**R. KAVITHA[1], D. CHITRA[2], N. K. PRIYADHARSINI[3], A. KALIAPPAN[4]**
*[1]Assistant Professor, Department of CSE, P. A. College of Engineering and Technology*
*[2]Professor and Head, Department of CSE, P. A. College of Engineering and Technology*
*[3]Assistant Professor, Department of CSE, P. A. College of Engineering and Technology*
*[4]Assistant Professor, Department of CSE, P. A. College of Engineering and Technology*

## Abstract

*Video event recognition aims to recognize the spatiotemporal visual patterns of events from videos. In recent years, event recognition has attracted growing interest from both academia and industry. Recognizing events in surveillance videos is still quite challenging, largely due to the tremendous intra class variations of events caused by visual appearance differences, target motion variations, viewpoint change and temporal variability. The existing system designed an extreme learning machine and action recognition algorithm for generalized maximum clique problem in video event recognition. The implemented system designed an enhanced ensemble deep learning and swarm based segmentation framework for video event recognition. The presented ensemble framework in that not only decreases the information loss and overfitting problems caused by single models. Initially, a video frames are taken as an input and most salient information extract from it. The VLAD for feature encoding is utilized for feature encoding. The segmentation process is done with the help of Random Inertia Weight based Particle Swarm Optimization (RIWPSO) of successive frames are exploited for pattern matching in a simple feature space. Thereafter, an Ensemble Learning (EL) is developed based on the performance of each SVM and Elman Recurrent Neural Network (ERNN) classifier on each feature set. Thus the simulation results demonstrate the effectiveness of the implemented enhanced ensemble deep learning technique for video event recognition compare to the existing methods.*

## INTRODUCTION

An ensemble is a system that consists of multiple smaller models. The goal of an ensemble is to combine smaller trained models to create a more accurate system. The algorithms have been designed that deal with the way these smaller models contribute to the final decision [21]. Figure 1 illustrates the basic framework for a classifier ensemble. For each example, the predicted output of each of these networks $o_i$ in Figure 1 is combined to produce the output of the ensemble. Many researchers have demonstrated that an effective combining scheme is to simply average the predictions of the network. Combining the output of several classifiers is useful only if there is disagreement among them. Obviously, combining several identical classifiers produces no gain. The popular ensemble methods are bagging, boosting and stacking.

 https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

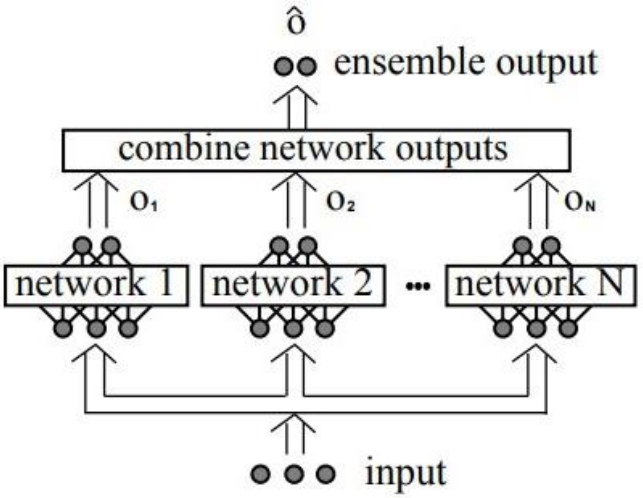


**Figure 1 A Classifier Ensemble of Neural Networks**

1. Bagging: Each member of the ensemble is generated by a different data set. It is good for unstable models. Where small differences in the input data set yield big differences in output. Many approaches have been developed using bagging ensembles to deal with class imbalance problems due to its simplicity and good generalization ability. The hybridization of bagging and data pre processing techniques is usually simpler than integration in boosting. A bagging algorithm does not require to recomputed any kind of weights, neither is necessary to adapt the weight update formula nor to change computations in the algorithm. Figure 2 shows Bagging.

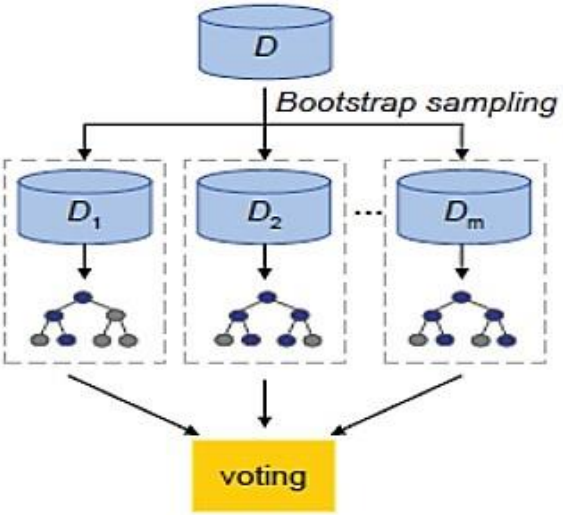


**Figure 2 Bagging**

2. Boosting: It is a family of ensemble learners. Its Basic idea is Weight the individual instances of the data set. It iteratively learns models and record errors and distribute the effort of the next round on the miss classified examples. The quantity of focus is measured by a weight, that initially is equal for all instances. After each iteration, the weights of misclassified instances are increased, on the contrary the weights of correctly classified instances are decreased. Figure 3 shows Boosting.


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

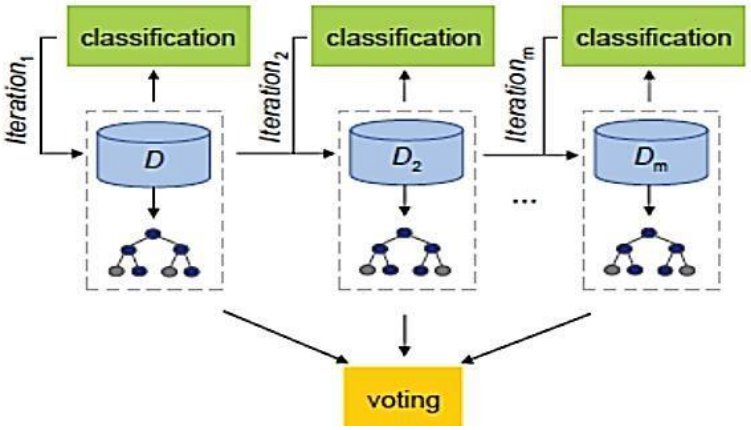


Figure 3 Boosting

3. Stacking: Its Basic idea is to have the output of a layer of classifiers as input to another layer. Stacking sometimes called stacked generalization involves training a learning algorithm to combine the predictions of several other learning algorithms. First, all of the other algorithms are trained using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs.

## PRINCIPLE OF PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization is an algorithm capable of optimizing a nonlinear and multidimensional problem usually reaches good solutions efficiently while requiring minimal parameterization [18] The algorithm and its concept of Particle Swarm Optimization (PSO) were introduced by James Kennedy and Russel Ebhart in 1995 [9] Its origins go further backwards since the basic principle of optimization by swarm is inspired in previous attempts at reproducing observed behaviors of animals in natural habitat, such as bird flocking or fish schooling and thus ultimately its origins are nature itself. These roots in natural processes of swarms lead to the categorization of the algorithm as one of Swarm Intelligence and Artificial Life. The basic concept of the algorithm is to create a swarm of particles move in the space around the problem space searching for goal or the place best suit needs given by a fitness function. A nature analogy with birds is the following: a bird flock flies in its environment looking for the best place to rest the best place can be a combination of characteristics like space for all the flock, food access, water access or any other relevant characteristic. Based on this simple concept there are two main ideas behind its optimization properties.

A single particle can be seen as a potential solution to the problem can determine “how good” its current position is. It benefits not only from its problem space exploration knowledge but also from the knowledge obtained and shared by the other particles. A stochastic factor in each particles velocity makes them move through unknown problem space regions. This property combined with a good initial

 https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

distribution of the swarm enable an extensive exploration of the problem space and gives a very high chance of finding the best solutions efficiently. Since its initial development PSO has had an exponential increase in applications. Antennas, especially in its optimal control and array design. Aside from there are many others like failure correction and miniaturization. Distribution Networks, especially in restructuring and load dis patching in electricity networks. Image and Video, this area is the one with most documented works in a wide range of applications, some examples are: face detection and recognition, image segmentation, image retrieval, image fusion, microwave imaging, contrast enhancement, body posture tracking. Scheduling**,** especially focused are flow shop scheduling, task scheduling in distributed computer systems, job shop scheduling and holonic manufacturing systems. But other scheduling problems are addressed such as assembly, production, train and project. Power Systems and Plants, especially focused are power control and optimization. Other specific applications are: load forecasting, photovoltaic systems control and power loss minimization.

## RANDOM INERTIA WEIGHT BASED PARTICLE SWARM OPTIMIZATION

In ensemble learning and swarm based segmentation framework is designed for video event recognition. Initially, a video frames are taken as an input and extract most salient information from frames. After the completion of improved dense trajectories feature extraction, Vector of Locally Aggregated Descriptors (VLAD) is designed for Vector of Locally Aggregated Descriptors (VLAD) encoding. Then Random Inertia Weight based Particle Swarm Optimization (RIWPSO) algorithm is utilized for segmentation. Finally, Ensemble Learning (EL) which includes Support Vector Machine (SVM) and Elman Recurrent Neural Network (ERNN) is introduced for classification. Figure 4 shows the flow diagram of implemented work.

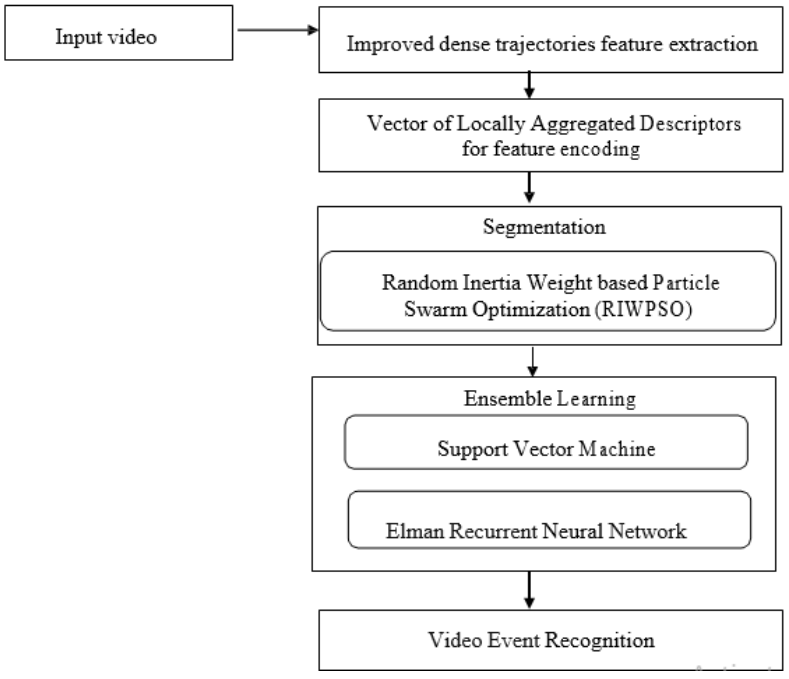


Figure 4. Flow Diagram of an Ensemble Learning and Swarm Based Segmentation


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

Initially, input videos are taken from VIRAT dataset. The implemented system presents a more effective approach of video representation using improved salient dense trajectories: first, detecting the motion salient region and extracting the dense trajectories by tracking interest points in each spatial scale separately and then refining the dense trajectories via the analysis of the motion saliency. Several descriptors trajectory displacement Computed HOG, HOF and MBH in the spatiotemporal volume aligned with the trajectories [32].

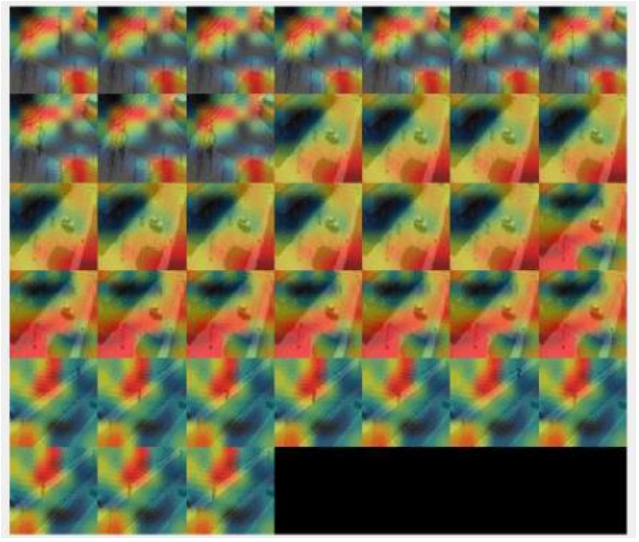

**Figure 5 Feature Map**

Figure 5 shows Feature Map after the feature extraction, VLAD encoding applied on improved trajectory, motion and appearance feature for global representation. The VLAD is an encoding and pooling technique. The VLAD is a type of global discriminative feature descriptor generated on a set of local features extracted from an image. Let X = $\{x_i\}_{i=1}^{N}$ be a set of local descriptors. A codebook C = $\{c_1, ..., c_k\}$ of k visual words can be learnt by the k-means algorithm. Each local descriptor $x_i$ can be quantized to the nearest visual word. For each visual word, the sum of the differences between the centre and each local descriptor assigned to this centre can be subsequently obtained. This can be expressed using Equation 1.

$$\delta_j(X) = \sum_{i=1}^{N} a_j^i (c_j - x_i) \quad (1)$$

Where, $a_j^i$ is a binary assignment weight indicating if the local descriptor belongs to this visual word and N is the number of local descriptors. The VLAD code is a concatenation vector of cumulated differences $\delta_j$ of each cluster is calculated using Equation (2).

$$v(X)= [\delta_1^T(X), \delta_2^T(X), \delta_3^T(x), \dots, \delta_k^T(X)] \quad (2)$$

The overall dimension of the VLAD code d×k, where d is the dimension of local descriptors and k is the number of dictionary entries.

 https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

In developed research work, segmentation is performed by using RIWPSO algorithm. The PSO is a nature inspired metaheuristic technique that simulates the behaviour of bird's flocking. In particle swarm optimization, every solution is represented as a discrete bird in the swarm, a particle in the search space. The canonical form of PSO consists of a population of particles, known as a swarm, with each member of the swarm being associated with position vector $x_i$ and velocity vector $v_i$. In developed research work, video frames are considered as particles [15]. The particles are initialized by a randomized position at the beginning of the search process and then after every iteration, the position and velocity of each particle is changed in such a way that it moves towards the desired $p_{best}$ and $g_{best}$ location. The accuracy is considered as fitness function. For each particle frames compute fitness function. If accuracy of the frames is higher than the best fitness value $p_{Best}$ in history set current value as the new $p_{Best}$. Choose the video frames with the best fitness of all the frames as the $g_{Best}$. The velocity frames are updated as per the Equation (3)[2].

$$V_i^{t+1} = \omega * V_i^t + c_1 * \text{rand } 1 * (PB_i - X_i^t) + c2 * \text{ rand } 2 * (gbest_i - X_i^t) \quad (3)$$

Where, i is the index of the frames, $i \in 1, \ldots, n$, $X_i^t$ is the position of frame i at iteration t, $V_i^t\ is$ the velocity of frame i at iteration t, rand1 and rand2 are two arbitrary numbers uniformly distributed in the range 0 and 1 and c1 and c2 are known as constants. The parameter ω is called inertia weight. Velocity is brought up to date dependent upon the inertia of prior velocity, experience of the particle itself as well as the neighboring particles. Particle position is brought up to date by using Equation (4).

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (4)$$

Where, $X_i^{t+1}$ -New position of the frame i, $V_i^{t+1}$-New velocity of the frame i

The inertia weight is one of the important parameters of PSO algorithm, whose choice is related to the balance between local and global search ability, affecting the convergence performance of algorithm, an appropriate value $\omega$ use the least number of iterations to find the optimal solution. Inertia weight value $\omega$ is determined by the Equation (5).

$$\omega = 0.5 + \frac{Rand()}{2} \quad (5)$$

The updated velocity of the frames are represented as Equation (6).

$$V_i^{t+1} = (0.5 + \frac{Rand()}{2}) * V_i^t + c_1 * \text{rand } 1 * (PB_i - X_i^t) + c2 * \text{ rand } 2 * (gbest_i - X_i^t) \quad (6)$$

 https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

RIWPSO Algorithm and Figure 6 shows feature particles of the event carrying object.

1. Initialize the particles (video frames)
2. Initialize Objective function: F(x)(accuracy)
3. Initialize velocity vector vi ;
4. Initialize position vector xi ;
5. Do until maximum number of iterations
6. For every particle i
7. Compute fitness function
8. If the accuracy is higher than the best value (pbest) in history then set current value as the new pbest
9. end For
10. Find the parameters with best accuracy value among the entire video frames and set it as gbest
11. Update $velocity$ ;
12. Update $position$ ;
13. segmentation
14. end

**Figure 6 Feature Particles of the Event Carrying Object**

## ENSEMBLE LEARNING

In implemented research work, an EL is developed based on the performance of each SVM and ERNN classifier on each feature set. The SVM was originally used for classification for binary classification on linearly separable data. The initial goal is to find optimal hyperplane. Hyperplane is boundary between two classes. The optimal hyperplane is not only separating two classes but also maximizes the margin between two classes. Margin is the longest distance between hyperplane and the nearest data (support vector) in


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

each class. Let $\{x_1, x_2, \dots, x_n\}$ is video frames in the dataset, $y_i \in \{-1,1\}$ is label of data, w is weighted vector. Hyperplane can be represented by Equation (7)[35]:

$$f(x) = wx_i + b = 0 \quad (7)$$

Hyperplane of each class can be represented as $y_i(wx_i + b) \geq 1, i = 1,2, \dots N$. Optimal margin is obtained from maximize the distance between hyperplane and support vector, $\frac{2}{||w||}$. Maximize optimal margin is equal to minimize $\frac{1}{2}||w||^2$. Classification of data class in SVM is obtained using Equation (8).

$$\min \frac{1}{2}||w||^2 \quad \text{subject to } y_i(x_i.w + b) \geq 1 \quad (8)$$

SVM was developed to solve problem for inseparable linear data. One of modification techniques to solve it is using kernel trick. In kernel trick, the data is projection into higher dimension such as the data is separable linearly. There are many types of kernels such as linear, polynomial, gaussian Radial Basis Function (RBF) and sigmoid function. In developed method, gaussian RBF kernel is used because RBF kernel has same performance as linear kernel on certain parameters. It has behaviour like sigmoid kernel function on certain parameters and the value interval [0,1] is small. Gaussian RBF kernel has been calculated using Equation (9).

$$k(x,x') = \exp\left(-\frac{||x-x'||^2}{2\sigma^2}\right) \quad (9)$$

Conducted event classification process using SVM is shown in Figure 7. First, the data will be divided into training dataset and testing dataset. In this method, video frames are partitioned randomly into k-part data where 1 part of data becomes testing dataset and k-1 part of data become training data. This is done as much as k times so that for each data is ensured to be training data and testing data. Training data is used in training process to find optimal hyperplane in SVM. After the training process is completed, tested the model using testing data and calculated the accuracy using Confusion Matrix.

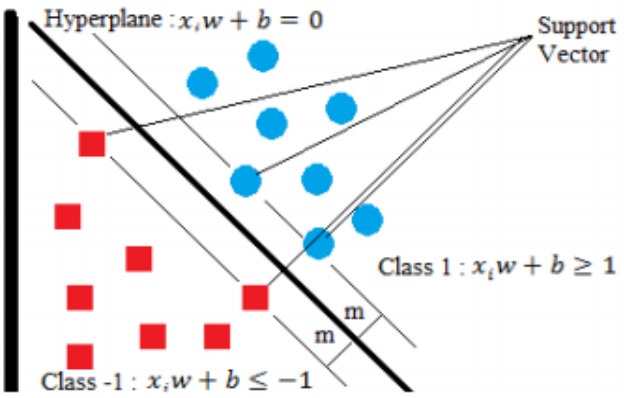


**Figure 7 Illustration of Classification in SVM**


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

The ERNN is a simple recurrent neural network. A recurrent network has some advantages, such as having time series and nonlinear prediction capabilities, faster convergence and more accurate mapping ability. The ERNN is composed of an input layer, a recurrent layer provides state information, a hidden layer and an output layer. Each layer contains one or more neurons propagate information from one layer to another by computing a nonlinear function of weighted sum of inputs [27].

A multi input ERNN model is exhibited, where the number of neurons in inputs layer is n and one output unit. Let $x_{it}$ (i=1,2,…,m) denote the set of input vector of neurons at time t , $y_{t+1}$ denotes the output of the network at time t+1 , $z_{jt}$ (j=1,2,...,n) denote the output of hidden layer neurons at time t and $u_{jt}$ (j=1,2,…,n) denote the recurrent layer neurons. $\omega_{ij}$ is the weight that connects the node i in the input layer neurons to the node j in the hidden layer. $c_j$ , $v_j$ are the weights that connect the node j in the hidden layer neurons to the node in the recurrent layer and output, respectively. Hidden layer stage is as follows, The inputs of all neurons in the hidden layer are expressed using Equation (10)

$$net_{jt}(k) = \sum_{i=1}^{n} \omega_{ij} x_{it}(k-1) + \sum_{j=1}^{m} c_j u_{jt}(k)) \qquad (10)$$

The outputs of hidden neurons are calculated by using Equation (11)

$$z_{jt}(k) = f_H\left(net_{jt}(k)\right) = f_H(\sum_{i=1}^{n} \omega_{ij} x_{it}(k) + \sum_{j=1}^{m} c_j u_{jt}(k)) \qquad (11)$$

The sigmoid function in hidden layer is selected as the activation function. The output of the hidden layer is computed using Equation (12),

$$y_{t+1}(\mathrm{k}) = f_T(\sum_{j=1}^{m} v_j z_{jt}(k)) \grave{} \qquad (12)$$

Where $f_T$ (x) is an identity map as the activation function. For a given input, the output probabilities from all SVM and ERNN are averaged before making a event classification. For output i, the average output $S_i$ is calculated using Equation (13).

$$S_i = \frac{1}{n}\sum_{j=1}^{n} r_j(i) \qquad (13)$$

Where $r_j$ (i) is the output i of network j for a given input pattern. The approach consists in applying a different weight for each network. In the validation set, networks that had a lower classification error will have a larger weight when combining the results. Given some input pattern, the output probabilities from all network are multiplied by a weight α before the prediction, it is represented using Equation (14).

$$S_i = \sum_{j=1}^{n} \alpha_j r_j(i) \qquad (14)$$

 https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

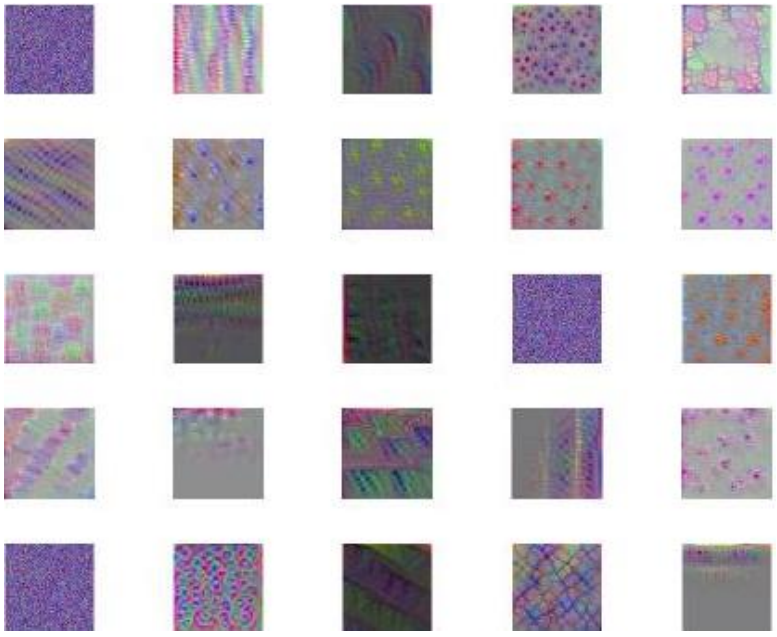

**Figure 8 Feature map in Recurrent Layer**

In developed research work, weighted mean is computed to calculate the weight α. The weight calculation is computed using Equation (15).

$$\alpha_k = \frac{A_k}{\sum_{i=1}^{n} A_i} \quad (15)$$

Where $A_k$ is accuracy in the validation set for the network k and i runs over the n. According to the average output of the SVM and ERNN network, the event classification is performed. In developed research work, EL which includes SVM and ERNN is utilized for video event recognition. The motion salient region is detected from input frames and the dense trajectories are extracted by tracking interest points in each spatial scale separately. Figure 8 shows feature map in recurrent layer. The video event recognition results are represented in Figure 9.


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

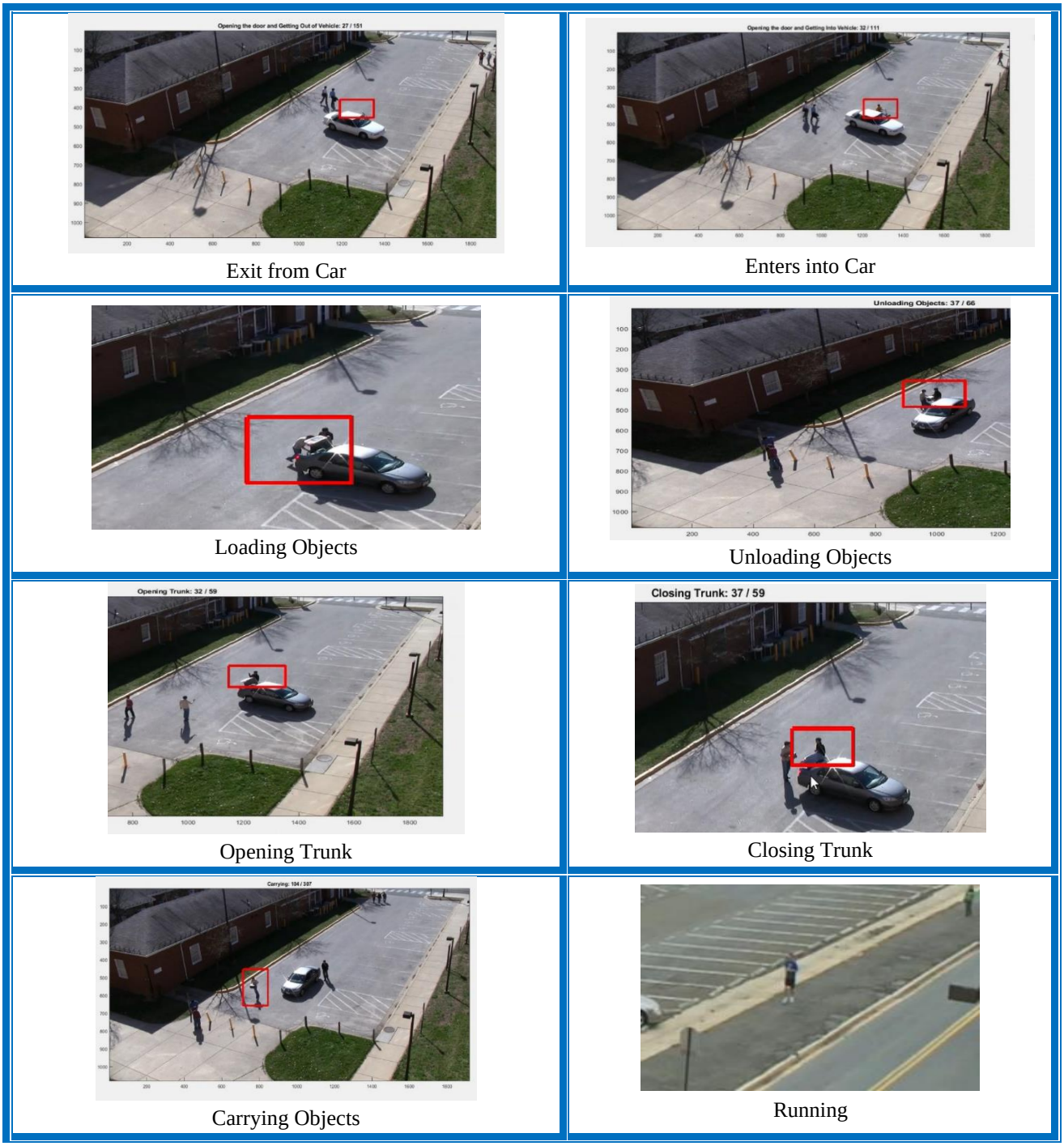


**Figure 9 Video Event Recognition**

## RESULTS AND DISCUSSION

In this work, the training videos are taken from VIRAT dataset videos with Resolution of 1080p X 720p dataset. MATLAB 2013a is used for implementation along processor of 3GB RAM. VLAD with EL takes 30 minutes for training and 12 seconds for testing. Here varying set of training videos are taken out which would have learned together to learn the different feature variation present among the videos of different kinds. The parameters being used Accuracy, F-Measure, Sensitivity, Specificity and Precision. The confusion matrix of VLAD with EL is shown in Figure 10.


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

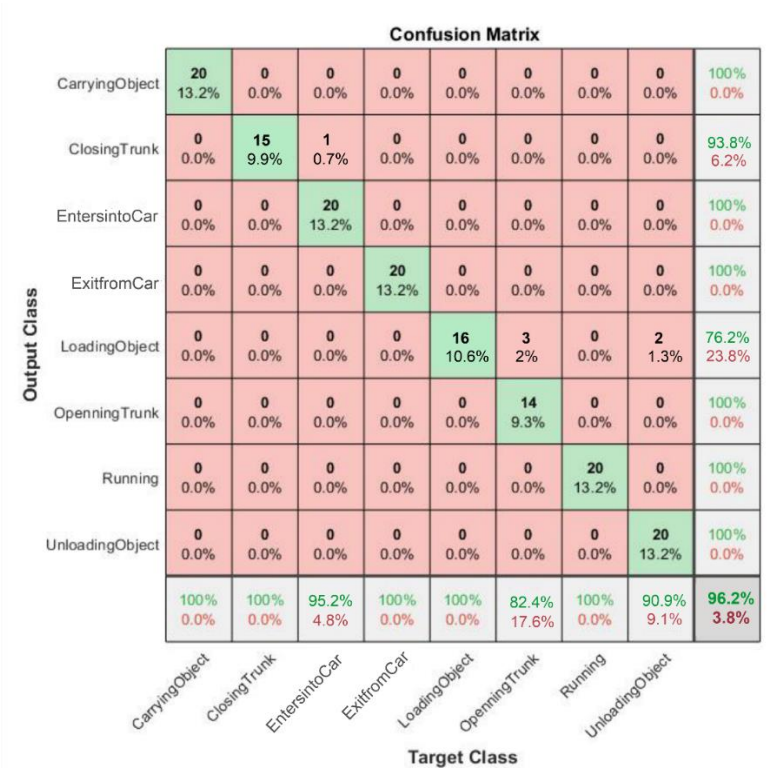


**Figure 10 Confusion Matrix for VLAD with EL**

Here varying set of training videos are taken out which would have learned together to learn the different feature variation present among the videos 8 Events. These videos would be learned features accurately based on which outcome would be made. Table 5.2 lists the performance metric Sensitivity, Accuracy, F-Measure, Specificity and Precision for the method IHDSM, VLAD with ELM and VLAD with EL.

**Table 0.1 Performance Comparison for 8 Events**

| Method | Parameters in % | | | | |
|---|---|---|---|---|---|
| | **Accuracy** | **Sensitivity** | **Specificity** | **Precision** | **F-Measure** |
| **IHDSM** | 90 | 90.17 | 92.87 | 91 | 91.5 |
| **VLAD with ELM** | 92.5 | 91.3 | 94.6 | 93.2 | 95.6 |
| **VLAD with EL** | 96.2 | 96.7 | 96.8 | 94.5 | 97.12 |

Accuracy of VLAD with EL approach is compared with IHDSM and VLAD with ELM approach which are shown in Figure 11. The experimental results shows that the VLAD with ensemble learning achieves 96.2% of accuracy.


https://doi.org/10.37896/pd91.4/91440

ISSN: 0369-8963

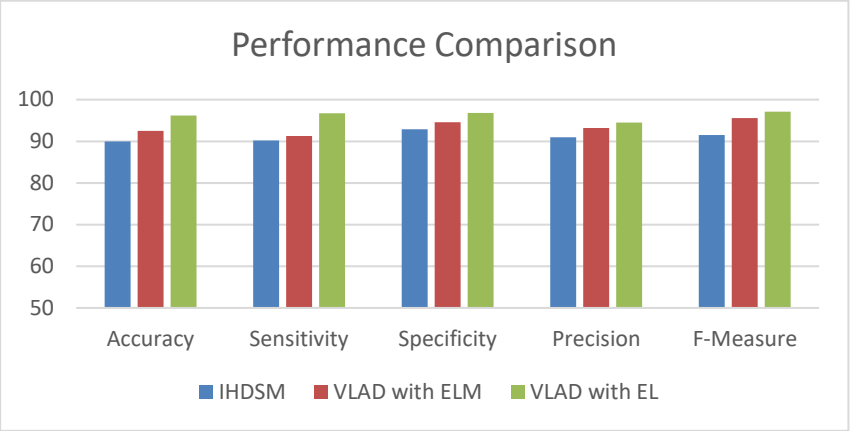


**Figure 11 Performance Comparison for 8 Events**

## CONCLUSION

The developed system designed ensemble learning and swarm based segmentation framework for video event recognition. In this work the segmentation process is performed by using Random Inertia Weight based Particle Swarm Optimization algorithm for pattern matching in a simple feature space. In order to improve the classification performance, Ensemble Learning is developed according to the performance Support Vector Machine and Elman Recurrent Neural Network classifier on each feature set. From the experimental results, it can be concluded that the Vector of Locally Aggregated Descriptors with ensemble approach achieves better performance compared with the existing system in terms of Accuracy, F-measure, Sensitivity, Specificity and precision.

## REFERENCES

1. Abdallah, ACB, Gouiffès, M & Lacassagne, L 2016, 'A modular system for global and local abnormal event detection and categorization in videos', Machine Vision and Applications, vol. 27, no. 4, pp. 463-481.

2. Amraee, S, Vafaei, A, Jamshidi, K & Adibi, P 2018, 'Abnormal event detection in crowded scenes using one-class SVM', Signal, Image and Video Processing, vol. 12, no. 6, pp. 1115-1123.

3. Arunnehru, J, Chamundeeswari, G & Bharathi, SP 2018, 'Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos', Procedia computer science, vol. 133, pp. 471-477.

4. Babiker, M, Khalifa, OO, Htike, KK, Hassan, A & Zaharadeen, M 2017, 'Automated daily human activity recognition for video surveillance using neural network', IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), Putrajaya; Malaysia; 28-30 November, pp. 1-5.

5. Bhatt, J & Patel, NS 2014, 'A Survey on One Class Classification using Ensembles Method', IJIRST-International Journal for Innovative Research in Science and Technology, vol. 1, no. 7, pp. 19-23.

 https://doi.org/10.37896/pd91.4/91440

6. Bouindour, S, Hittawe, MM, Mahfouz, S & Snoussi, H 2017, 'Abnormal event detection using convolutional neural networks and 1-class SVM classifier', 8th International Conference on Imaging for Crime Detection and Prevention (ICDP), Madrid, Spain 13 – 15 December, pp. 1-6.

7. Coşar, S, Donatiello, G, Bogorny, V, Garate, C, Alvares, LO & Brémond, F 2016, 'Toward abnormal trajectory and event detection in video surveillance', IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 3, pp. 683-695.

8. Doshi, K & Yilmaz, Y 2020, 'Continual Learning for Anomaly Detection in Surveillance Videos', Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, June 14 2020 to June 19, pp. 254-255.

9. Feng, Y, Yuan, Y & Lu, X 2017, 'Learning deep event models for crowd anomaly detection', Neurocomputing, vol. 219, pp. 548-556.

10. Gan, C, Wang, N, Yang, Y, Yeung, DY & Hauptmann, AG 2015, 'Devnet: A deep event network for multimedia event detection and evidence recounting', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7 2015 to June 12, pp. 2568-2577.

11. Hubel, DH & Wiesel, TN 1962, 'Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex', The Journal of Physiology, vol. 160, pp. 106–154.

12. Ionescu, RT, Smeureanu, S, Popescu, M & Alexe, B 2018, 'Detecting abnormal events in video using Narrowed Motion Clusters', Computer Vision and Pattern Recognition, pp. 1-17.

13. Javanbakhti, S & Zinger, S 2012, 'Fast abnormal event detection from video surveillance', Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), Nevada, United States of America, 16th to 19th July.

14. Kaliappan. A & Chitra. D," Kringing Regressive Map reduce Entropy Feature Extraction based Rocchio Adaptive Boost Ensemble Classifier for Early Disease Diagnosis with Big Data", Journal of Dynamic Systems and Applications, Dynamic Publishers Inc., US, vol.30, no. 6, pp. 964-980.

15. Kaliappan. A & Chitra. D," Analysis of Big Data Analytics in Healthcare sector: Applications and Tools", Journal of Computational and Theoretical Nanoscience, American Scientific Publishers, US, vol.17, no.12, pp. 5605-5612(8).

16. Kaliappan. A, Dineshkumar. T, Sabarivel. M, "Hierarchical Kriging DNN Model for Epileptic Seizure Detection from EEG Signals", TNSCST Sponsored 8th International Conference on Latest Trends in Science, Engineering and Technology (ICLTSET'22) organized by Karpagam Institute of Technology, Coimbatore.

17. Karaulova, IA, Hall, PM & Marshall, AD 2002, 'Tracking people in three dimensions using a hierarchical model of dynamics', Image and Vision Computing, vol. 20, n0 9-10, pp. 691-700.

 https://doi.org/10.37896/pd91.4/91440

18. R. Kavitha and D. Chitra, "An improved hybridized deep structured model for accurate video event recognition" in the Journal of Ambient Intelligence and Humanized Computing, Springer - Verlag GmbH Germany, part of Springer Nature, June 2020, Volume 12, ISSN 1868-5137.

19. R. Kavitha, D. Chitra and N. K. Priyadharsini, "Video Event Recognition Using Conditional Random Fields" in the International Journal Annals of the Romanian Society for Cell Biology, ISSN:1583-6258, Vol. 25, Issue 4, 2021, Pages. 6565 – 6573. Scopus Indexed.

20. R. Kavitha and D. Chitra, "An Extreme Learning Machine and Action Recognition Algorithm for Generalized Maximum Clique Problem in Video Event Recognition" in the Journal of Dynamic Systems and Applications, Volume no 30, Issue no 8, 2021, ISSN 2693-5295, page no 1228 – 1249

21. Kwon, J & Lee, KM 2014, 'A unified framework for event summarization and rare event detection from multiple views', IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1737-1750.

22. Lai, KT, Yu, FX, Chen, MS & Chang, SF 2014, 'Video event detection by inferring temporal instance labels', Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, USA, 23 to 28 June, pp. 2243-2250.

23. Medel, JR & Savakis, A 2016, 'Anomaly detection in video using predictive convolutional long short-term memory networks', Computer Vision and Pattern Recognition, pp. 1-27.

24. Oh, S, Hoogs, A, Perera, A, Cuntoor, N, Chen, CC, Lee, JT & Swears, E 2011, 'A large-scale benchmark dataset for event recognition in surveillance video', IEEE Conference on CVPR, Colorado Springs, CO, USA, 20-25 June, pp. 3153-3160.

25. Priyadharsini, N, K, & Chitra, D, 2021, 'A machine learning and swarm intelligence based approaches for anomaly detection in extremely crowded scenes', Journal of Dynamic Systems and Applications, vol. 30, no. 8, pp. 1250 – 1272, doi.org/10.46719/ dsa20213083.

26. Priyadharsini.N.K ,D. Chitra , A kernel support vector machine based anomaly detection using spatio-temporal motion pattern models in extremely crowded scenes, Springer, Journal of Ambient Intelligence and Humanized Computing, Doi.org/10.1007/s12652-020-02000-3.

27. Selvi, V & Umarani, R 2010, 'Comparative analysis of ant colony and particle swarm optimization techniques', International Journal of Computer Applications, vol. 5, no. 4, pp. 1-6.

28. Shafiee, MJ, Azimifar, Z & Wong, A 2015, 'A deep-structured conditional random field model for object silhouette tracking', PloS one, vol. 10, no. 8, pp. 1-17.

29. Tavassolipour, M, Karimian, M & Kasaei, S 2013, 'Event detection and summarization in soccer videos using bayesian network and copula', IEEE Transactions on circuits and systems for video technology, vol. 24, no. 2, pp. 291-304.

ISSN: 0369-8963

30. Vu, TH & Wang, JC 2016, 'Acoustic scene and event recognition using recurrent neural networks', Detection and Classification of Acoustic Scenes and Events, pp. 1-3.

31. Wang, D, Tan, D & Liu, L 2018, 'Particle swarm optimization algorithm: an overview', Soft Computing, vol. 22, no. 2, pp. 387-408.

32. Wang, H, Kläser, A, Schmid, C & Liu, CL 2013, 'Dense trajectories and motion boundary descriptors for action recognition', International journal of computer vision, vol. 103, no. 1, pp. 60-79.

33. Wu, X & Jia, Y 2012, 'View-invariant action recognition using latent kernelized structural SVM', European conference on computer vision, Florence, Italy, October 7-13, pp. 411-424.

34. Xu, D, Yan, Y, Ricci, E & Sebe, N 2017, 'Detecting anomalous events in videos by learning deep representations of appearance and motion', Computer Vision and Image Understanding, vol. 156, pp. 117-127.

35. Zhongliang, F 2013, 'A universal ensemble learning algorithm', Journal of Computer Research and Development. vol 53, no. 4, pp. 149-158.

 https://doi.org/10.37896/pd91.4/91440