

Human activity recognition in egocentric video using HOG, GiST and color features

K. P. Sanal Kumar¹ · R. Bhavani²

Received: 20 March 2018 / Revised: 30 March 2018 / Accepted: 20 April 2018 /
Published online: 7 May 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract With the rapid increase in digital technology, most research areas are involved in human activity recognition, which can help to analyze the activities of patients. A novel approach for human activity recognition in egocentric video has been invoked in this research article. Generally, only the objects are identified, but the actions are not recognized. With this motivation and new trends, this paper presents an efficient technique to recognize the activities. In our approach, first the various activity dataset is trained, and the feature vector values are stored for various activities, which are applied to the testing inputs. Here, we use a filtering technique, i.e., a median filter followed by a segmentation method using watershed and feature extraction, such as a Histogram of Oriented Gradient (HOG), Color and GiST and a combination of all Features. Features are reduced using a genetic algorithm, and classification is done using Support Vector Machine (SVM) and a Random Forest classifier. The experimental results demonstrate that the Random Forest classifier outperformed the SVM classifier.

Keywords Computer vision · egocentric vision · Histogram of Oriented Gradients · Genetic algorithm · Random Forest classifier

1 Introduction

Human action is one of the main topics in the recognition of computer vision. Other computer vision uses the motion boundaries, e.g., object segmentation in videos [7, 23], for action recognition. Human activity detection is a tough task due to variations in the shape of the body and pose, etc. It is even harder to find in dark backgrounds or moving cameras. Videos

✉ K. P. Sanal Kumar
sanalprabha@yahoo.co.in

R. Bhavani
bhavaniaucse@gmail.com

¹ Department of ECE, Annamalai University, Chidambaram, India

² Department of CSE, Annamalai University, Chidambaram, India

containing human actions or activity convey the essential meaning for better understanding the scenario. Motion detection or estimation is the effective information for action recognition. Analyzing the activities of a person can help elderly people, people who are disabled, patients and so on. The motivation is to develop a system for patient monitoring.

However, the recognition of actions in unconstrained digital videos has been a challenging problem in computer vision technology. The prime factor in action recognition is a representation of an action video. Action video representation is based on the following circumstances: i) human pose: extracts the human information based on physical condition or structure [28], ii) action: captures the whole-body shape or appearance and motion information, iii) local features: extract valid space-time cuboids, and iv) unsupervised feature learning-based methods: learn the representation by hierarchical networks [14].

This paper proposes and combines motion-based feature sets for human activity detection in videos.

- i) Apply the filter technique and segmentation method in the image.
- ii) Evaluate them independently and in combination with the Color, GiST and Histogram of Oriented Gradient (HOG) appearance descriptors that were developed for human detection in videos.

The paper is structured as follows: after reviewing the methodologies in Section 2, we present our novel approach for filtering, segmentation methods, feature extraction, feature reduction and classification in Section 3. We then discuss the datasets used in our experiments and the result evaluation in Section 4. The multimodal egocentric activity dataset is used in this research work.

2 Literature survey

Recorded egocentric images and videos of daily activities from wearable cameras are important to assist memory recollection for both memory-impaired and unimpaired persons. Since egocentric images and videos about daily activities are long and unstructured, the ability to retrieve past egocentric images and videos could support and augment human memory. The current egocentric image and video retrieval methods use manually and automatically labeled [6, 8] images as user queries. However, these approaches let users who need memory support describe what they forgot in their own words as user queries similar to the past visual experience of what users want to remember. Furthermore, most previous methods have ignored the valuable feature of human-physical world interactions, which usually associate our daily activities and visual experience. [4] proposed egocentric vision-based activity recognition methods using gaze location, segmented hand region and object detection methods by [18] as features of human activities. These approaches mainly classify egocentric videos using supervised techniques that require manually labeled training data, so that predictable activities are limited. [10] proposed human activity recognition, and the recognition procedure is implemented in 3 steps, i.e., feature extraction, evaluation, and classification, using a Feed Forward Neural Network (FFNN) classifier. [13] proposed a wearable device, which was based on a sensor called e-Watch, and used it to test human activity recognition and location. [24] proposed the HOF and MBH descriptors and a combined descriptor based on Dataset (LENA). [16] experimented with a framework to segment and arrange a set of egocentric videos using a convolutional neural network. [29] proposed

multiple deep learning pipelines to study the appearance and motion patterns that can predict the activity of the wearer. [27] identified the associations between objects with some scenes to enhance the object detection based on the scene content. [3] recognized the objects using the deep convolutional neural network. [5] implemented swarm search in the context of human activity recognition. [12] experimented with active and inactive manners in activity recognition. [21] identified the activities in egocentric videos.

3 Proposed work

Our proposed work is to find the human activity and is based on the watershed segmentation algorithm and three feature extraction techniques. The genetic algorithm is used for feature reduction and finally SVM and Random Forest classifiers are used to find the type of activity based on training and testing. First, the input videos with a frame rate of 29 frames/s are converted into frames of size 64*64, and then a median filter is applied to remove noise. Here, the databases are divided into two levels, i.e., top level and second level. Once the noise is removed, then segmentation is applied by watershed. The main purpose of image segmentation is to divide the image into meaningful structures. We used three feature extraction methods in the design to find the best combination along with the segmentation for activity detection. The HOG, Color and GiST feature extraction methods are used in the design. The genetic algorithm is used for feature reduction, and the SVM and Random Forest classifiers are used to find the activity type. Figure 1 depicts the proposed model.

3.1 Filtering

The purpose of applying filtering is to enhance the quality of an image. Median filtering is nonlinear. [26] introduced the use of the median filter in signal processing. The filter process is based on moving the image, pixel by pixel and then replacing each value with the median value of the neighboring pixels. The neighboring pattern is called a “window”, which slides, pixel by pixel, over the entire image. First sorting the pixels calculates the median of all the values from a window and then replaces the pixel being considered with the middle value. Figure 2 depicts the median filtering method. A window of size three is used.

3.2 Watershed segmentation

Watershed-based image segmentation algorithms consist of constructing a symbolic representation of the image. Watershed algorithms mainly have two classes. The first class implements the flooding-based watershed algorithms, which is a traditional approach, whereas the second class contains rain falling-based watershed algorithms. The connected components-based watershed algorithm [1] provided very good performance compared to all other algorithms, and it falls under the rain falling-based watershed algorithm approach. It provides very good segmentation results and needs less computational complexity for implementation. Figure 3 shows the implementation of the watershed algorithm. The algorithm follows the below steps:

1. Calculation of image gradient value
2. Algorithm for watershed segmentation procedure
3. Merging procedures

III. PROPOSED WORK

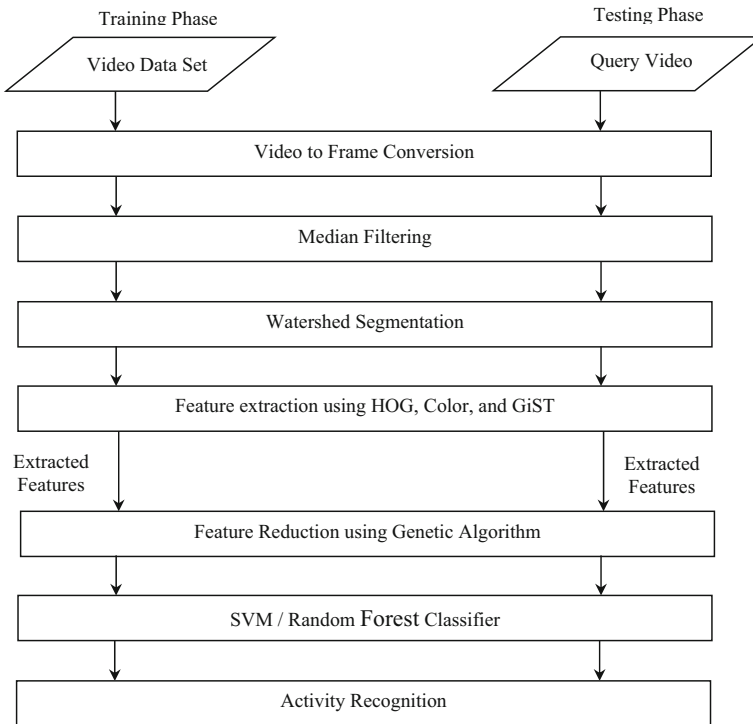


Fig. 1 Block diagram of proposed system

3.2.1 Gradient calculations

The watershed transforms are calculated [19] based on the image gradient. The first partial derivative of an image is defined as a gradient. $G(x, y)$ represents the gradient values of the

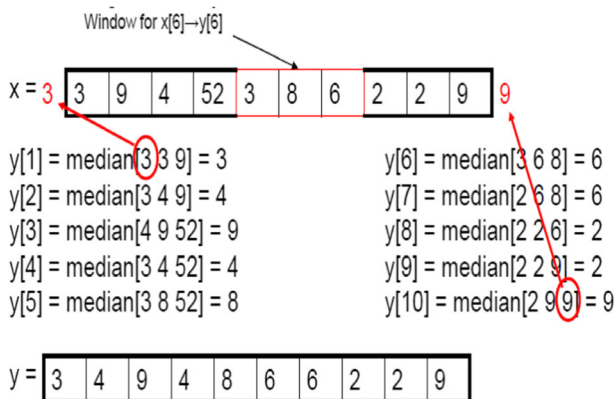


Fig. 2 An approach to the median filter

initial segmented image, obtained by a gradient operator approximation in x-y directions as two masks of 3*3.

$$\begin{aligned} U_x(i, j) &= (2 + 4c)^{-1} \{u(i + 1, j) - u(i - 1, j) + c[i + 1, j + 1) - u(i - 1, j + 1) + u(i + 1, j - 1) - u(i - 1, j - 1)]\} \\ U_y(i, j) &= (2 + 4c)^{-1} \{u(i, j + 1) - u(i, j - 1) + c[u(i + 1, j + 1) - u(i - 1, j - 1) + u(i - 1, j + 1) - u(i - 1, j - 1)]\} \end{aligned} \quad (1)$$

$$G(x, y) = \sqrt{(\partial f / \partial x)^2 + (\partial f / \partial y)^2} \quad (2)$$

where $c = (\sqrt{2}-1) / (2-\sqrt{2})$. $G(x, y)$ is calculated from the gradient image, and the gradient values on the border of the input image are the same as its inner pixels.

3.2.2 Algorithm procedures

The three main processes are indicated in Fig. 3. Pre-processing comes at the first stage, image segmentation based on the watershed algorithm comes at the second stage and post-processing is the last stage. The input images are pre-processed first and then given as a second stage to watershed-based segmentation. The final stage of the segmented image is the result after the post-processing stage. The pre-processing and post-processing stages are important to overcome the problem of over-segmentation in image segmentation.

Watershed segmentation is applied to the gradient of an image, where the regions in the image characterized by small variations in gray levels have small gradient values, rather than to the image itself. The watershed transform function is based on finding the high-intensity gradients (watersheds) that divide the neighboring local minima (basins). The watershed line pixels are obtained by a marker image, which includes zero marker values. We used a 3*3 mask to scan this image to find the zero values and convert them to their intensity values as in the original image. Comparing these values with their neighboring pixels, intensity is assigned to one marker region. All zero marker values (watershed pixels) are deleted to obtain a second marker image that represents the markers of the image regions only.

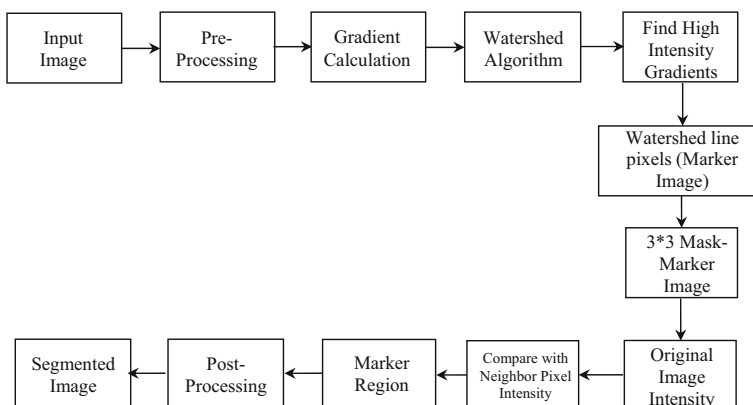


Fig. 3 Block diagram of watershed-based image segmentation

3.2.3 Merging procedures

The number of pixels of each region in the image (N_i) is calculated by using a marker image and then by finding the mean intensity value (μ) of each region (i), using eq. (3).

$$(\mu_i) = \frac{\sum_{N \in i} \text{origion pixels intensity of (i)}}{N_i} \quad (3)$$

From the original input image, the intensity values of region i are obtained because their positions in the two images are the same. The merging procedure is based on i) merge the pair of regions and ii) edge strength.

3.3 HOG feature extraction

Histogram of Oriented Gradient (HOG) descriptors [2] are feature descriptors that use the direction of intensity of the gradients and edge directions. Figure 4 shows the flow chart to extract the HOG descriptor.

The HOG descriptor is divided into multiple steps:

Computing gradient: We first calculate the gradient values for all the pixels in the image using any derivative mask over the image in the horizontal and vertical directions. Some common derivative masks are the Sobel operator and the Prewitt operator, among others, but the original algorithm recommends that you use a 1D derivative mask, that is, $[-1, 0, +1]$.

Orientation binning: Create a histogram of the weighted gradients that were computed in the previous step. The gradient values are divided into bin values, ranging from 0 to 180 or from 0 to 360 (depending on whether we are using signed or unsigned gradient values).

Combining cells to form blocks: After computing histograms for each cell, we combine these cells into blocks and form a combined histogram of the block using its constituent cells' normalized histograms. The final HOG descriptor is a vector of the normalized histograms. Here, 8102 features are extracted [20, 22].

Fig. 4 Flowchart of an HOG descriptor

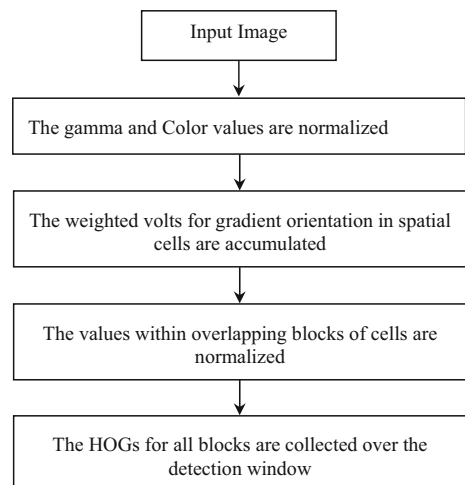


Fig. 5 Reference image

Building the classifier: In the final step of the algorithm, feed the HOG feature vectors that were computed in the previous step into your favorite learning algorithm, and build a model that will later be used to detect objects in images.

3.4 Color feature

At present, there is an increase in the data of a color image [30]. Because the amount of information in color images is larger, there is widespread concern about their use. Some acquired data from color images, such as ionosphere, aurora, geomagnetism, biology, ocean, and meteorological data, need to carry out the effective feature extraction, fine classification. These issues require the development of feature extraction methods or algorithms in color images for edges, corners, etc.

For the reference image shown in Fig. 5, the RGB image average values that represent RGB in an image are given as $H = \{134.2338, 101.3403, 87.1001\}$.

The reference image is divided into 16 blocks (4*4). We obtain 64 color values from each block, which show the color in that block only. In total, there are 1024 values for the full image. The color moment method is used, which is based on color distribution that is calculated as a probability distribution. The probability distributions are categorized by many unique moments, which are given below:

- 1) Mean represents average color value
- 2) Standard deviation is a function of the square root of the variance.
- 3) Skewness.

Figure 6 shows 9 moments of the reference image (Fig. 5).

The columns correspond to each of our channels, and the rows, to moments. The moment's values provide the color similarity between the images. The total value of the weighted differences defines the similarity function between image distributions. An integrated approach of the color feature method is used to obtain the accurate output from a video. By this method, the features in various classes are converted into one feature. This compares database images with respect to the query image. Here, 9 features are extracted from each frame.

Fig. 6 Reference image color moments

$$\begin{bmatrix} 0.5224 & 0.40519 & 0.3376 \\ 0.0862 & 0.0881 & 0.0915 \\ -0.0789 & -0.0733 & -0.0806 \end{bmatrix}$$

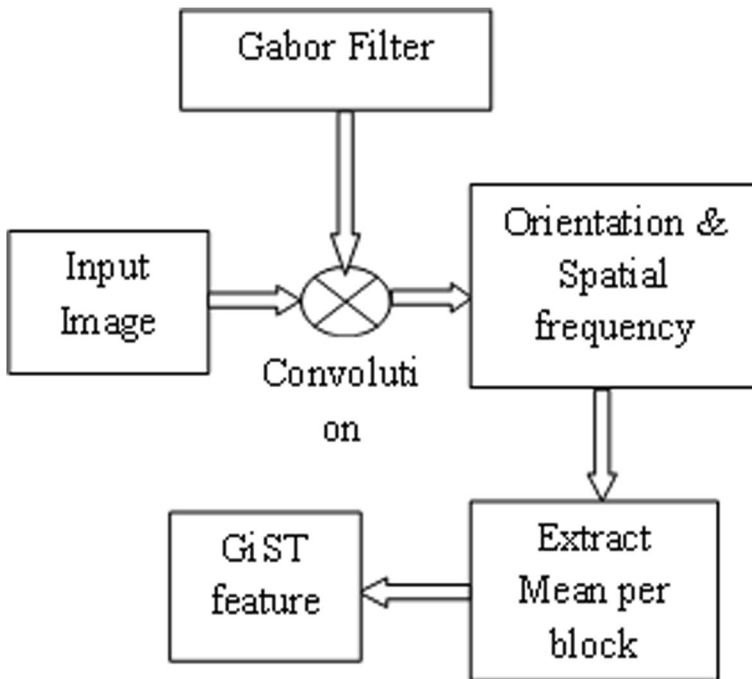
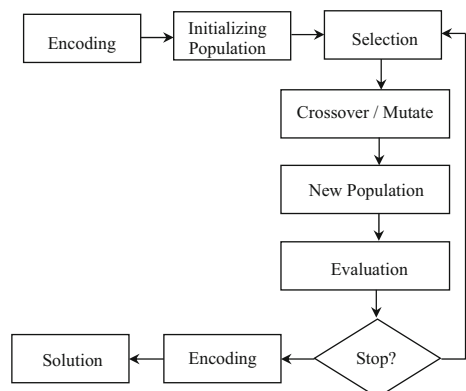


Fig. 7 GiST feature-extracting process

3.5 GiST feature

GiST feature extraction [17] is based on the convolution process and mean per block calculation. This process is based on a filtering algorithm (the Gabor filter) with an orientation for the convolution process, different spatial frequency and mean calculation by splitting the image into several small block sizes, i.e., an 8*8 block. The convolution process done is in the Fourier domain for computing

Fig. 8 Genetic algorithm flowchart



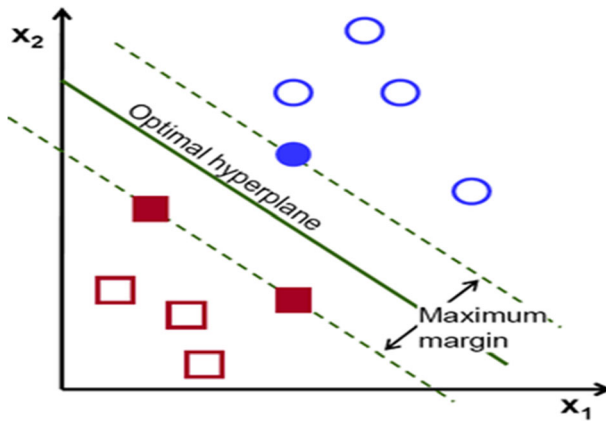


Fig. 9 Support vector machine

efficiency and switches back to the time domain for mean average per block calculation. Figure 7 shows the extracting process of GiST. In our research, 512 features are extracted.

3.6 HOG+GiST+Color feature

In this research work, the features are combined together to form a single feature. Therefore, a total of 8623 features are extracted.

3.7 Genetic algorithm

The genetic algorithm (GA) [9] is one prospective option for feature selection. In any typical GA optimizer, an initial population is created with a predetermined number of strings, also

Fig. 10 Grouping activity level

| | | |
|-------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Activities | Ambulation | Riding on Elevator Down Riding on Elevator Up Riding on Escalator Down Riding on Escalator Down Sitting Walking Walking Downstairs Walking Upstairs |
| | Daily Activities | Drinking Eating Making Phone Calls Texting/Talk on the Phone |
| | Exercise | Cycling Doing Push Up Doing Sit Up Running |
| | Office Work | Organizing Files Reading Working on PC Writing Sentences |
| | First Level | Second Level |



Fig. 11 Sample frames from multimodal egocentric dataset

called chromosomes. Each of these represents an individual. The set of individuals forms the current generation. A fitness value is associated with every chromosome. The choice of the fitness function totally depends on the nature of the problem. The performance of the GA leads to the existence of three phases in a typical GA optimization. The three phases in the GA optimization process are (1) generation of the initial population (2) reproduction and (3) generation replacement. Each individual in the generation is assigned a fitness value by evaluating the fitness function. In the reproduction step, a new generation is formed from the current generation. In this process, pairs of individuals are chosen to act as parents. The selection may be based on the fitness function. Crossover and mutation are performed on the parent chromosomes, and a population of children is produced. In the crossover process, a node is selected randomly in each pair of parent chromosomes. Then, the two parts of the chromosomes are exchanged to form two new chromosomes. In the mutation process, a bit is randomly selected in a chromosome, and the value is changed (in binary GA a '0' will become '1' and vice versa). Then, a new generation is formed with these children. The above operations are repeated until the new generation is filled. The generic flowchart of a GA system is shown in Fig. 8. After reduction, the HOG feature was reduced to 800, GiST was reduced to 256 and Color moments were reduced to 6.

3.8 Random forest classifier

The Random Forest classifier algorithm [11] is based on regression and classification. A Random Forest classifier is a collection of tree predictors that is called a forest. A Random Forest classifier works by first taking the feature vector as input and classifies with every tree

Table 1 Performance metrics of first level categories

| Features | Sensitivity | Specificity | Accuracy |
|----------------|-------------|-------------|----------|
| HOG | 82.4 | 80.7 | 81.6 |
| GiST | 81.5 | 80.4 | 80.9 |
| Color | 82.1 | 81.8 | 81.9 |
| HOG+GiST+Color | 84.3 | 82.9 | 83.6 |

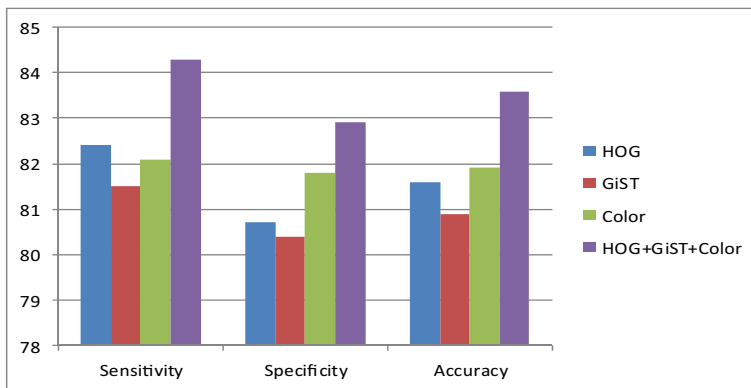


Fig. 12 Evaluation for first-level categories using SVM

in the forest. This outputs the class label that obtained the maximum “votes”. The classifier response is the average of the responses obtained from all the trees in the forest, in the case of regression. Training of all trees is done with common parameters, but the training is done on different sets. A bootstrap procedure is used to generate the original training set. A random selection process is done for each set of training with the same vector number as in the original set, which is represented as (N) . The vectors are chosen with replacement. Some vectors are absent, and some will occur more than once. The variables that are used to find the best split are not possible with all variables at each node of each trained tree but a subset. A new subset is created at each node. But its size is fixed. It is considered as a training parameter set to the square root of the number of variables. No accuracy estimation procedures such as bootstrap or cross-validation are needed in a Random Forest. The estimation of error is done internally during the training process. The output of the Random Forest is to recognize activity for 4 top-level categories and 20 s-level categories.

3.9 Support vector machine

Support vector machines represent systems that are associated with learning algorithms for the classification of values. An SVM algorithm develops a system that assigns examples to categories [25]. New data are trained to fit the categories, which are divided already and are predicted. SVMs can perform efficiently as a nonlinear classifier. Here, multiclass SVM is used.

Figure 9 shows the separation of values with the help of the optimal hyperplane, which is cited from [15]. The SVM performance is based on the hyperplane that provides the highest minimum distance to the training values. The output of the SVM is to recognize activity for 4 top-level categories and 20 s-level categories.

Table 2 Performance metrics for Random Forest classifier

| Features | Sensitivity | Specificity | Accuracy |
|----------------|-------------|-------------|----------|
| HOG | 85.36 | 81.84 | 83.6 |
| GiST | 87.32 | 82.47 | 84.89 |
| Color | 83.12 | 82.83 | 82.97 |
| HOG+GiST+Color | 88.36 | 86.84 | 87.51 |

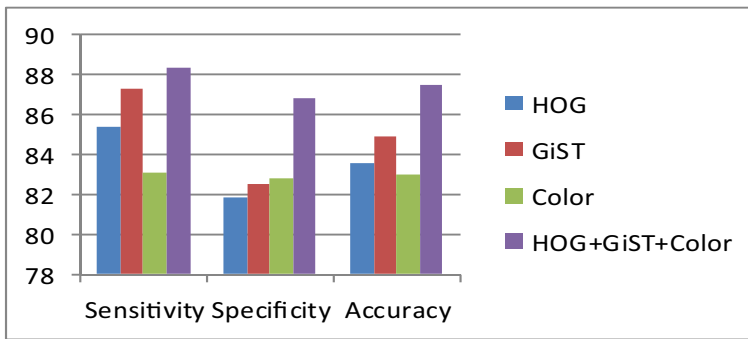


Fig. 13 Evaluation for first-level categories using random forest

4 Experimental details

The proposed method has been implemented in MATLAB 2015a in a Windows environment with a system configuration of Intel Pentium VII Generation I5 with 4 GB RAM. We evaluate the performance and comparison using different descriptors and segmentation methods. In this research article, multimodal egocentric activity dataset is used. Here, 1000 samples were taken. 10 cross fold validation is used where all samples were trained and tested. Four first-level categories and 20 s-level categories are applied as shown in Fig. 10.

4.1 Dataset collection

The dataset used in the proposed approach contains 20 activities recorded in four scenarios. Each scenario recorded two sets for each activity. Thus, every activity category has twenty clips. The duration will be approximately 20–30 s for every clip recorded video. The activity categories are Riding on Elevator Down, Riding on Elevator Up, Riding on Escalator Down, Riding on Escalator Down, Walking, Sitting, Walking Downstairs, Walking Upstairs, Drinking, Eating, Making Phone Calls, Texting, Cycling, Doing Push Up, Doing Sit Up, Running, Organizing Files, Reading, Working on PC, and Writing Sentences. Sample images are shown in Fig. 11.

4.2 Performance metrics used

Sensitivity is defined as the ratio between the number of true positives and the summation of true positives and false negatives. It is given as

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN})$$

Table 3 Performance metrics for SVM classifier

| Features | Sensitivity | Specificity | Accuracy |
|----------------|-------------|-------------|----------|
| HOG | 70.12 | 68.21 | 69.23 |
| GiST | 78.32 | 76.54 | 77.43 |
| Color | 66.54 | 63.21 | 64.88 |
| HOG+GiST+Color | 80.14 | 78.86 | 79.5 |

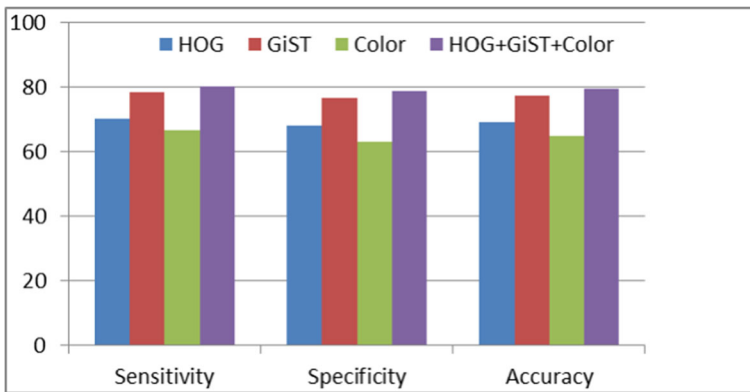


Fig. 14 Evaluation for second-level categories using SVM

Specificity is defined as the ratio between the number of true negatives and the summation of true negatives and false positives. It is given as

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP})$$

Accuracy is defined as the average of sensitivity and specificity. It is given as

$$\text{Accuracy} = (\text{Sensitivity} + \text{Specificity})/2$$

4.3 Evaluation for first-level categories using SVM

Table 1 and Fig. 12 depict the performance metrics of the SVM classifier for various metrics. The table and figure show that HOG+GiST+COLOR provided better results than other features for first-level categories using SVM.

4.4 Evaluation for first-level categories using random forest

Table 2 and Fig. 13 depict the performance metrics of the Random Forest classifier for various metrics. The table and figure show that HOG+GiST+COLOR provided better results than other features for first-level categories using Random Forest.

Table 4 Performance metrics for random forest classifier

| Features | Sensitivity | Specificity | Accuracy |
|----------------|-------------|-------------|----------|
| HOG | 78.32 | 72.11 | 75.21 |
| GiST | 83.45 | 77.59 | 80.52 |
| Color | 68.54 | 65.21 | 66.88 |
| HOG+GiST+Color | 84.14 | 80.86 | 82.45 |

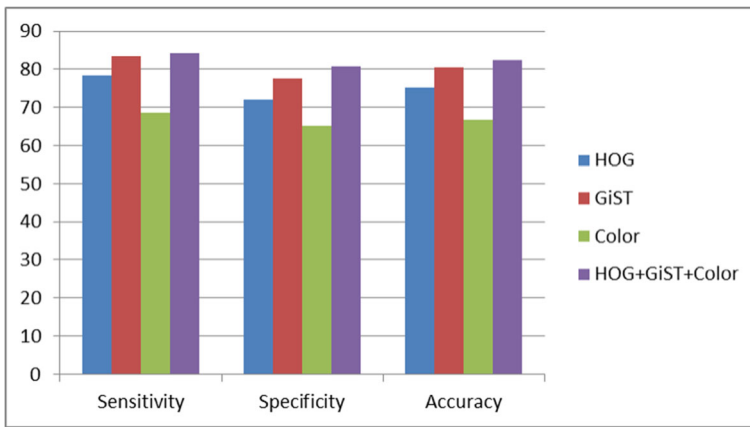


Fig. 15 Second-level activities comparison between different features using random forest

4.5 Evaluation for second-level categories using SVM

Table 3 and Fig. 14 depict the performance metrics of the SVM classifier for various metrics. The table and figure show that HOG+GiST+COLOR provided better results than other features for first-level categories using SVM.

4.6 Evaluation for second-level categories using Random Forest

Table 4 and Fig. 15 depict the performance metrics of the Random Forest classifier for various metrics. The table and figure show that HOG+GiST+COLOR provided better results than other features for second-level categories using the Random Forest classifier (Figs. 16 and 17).

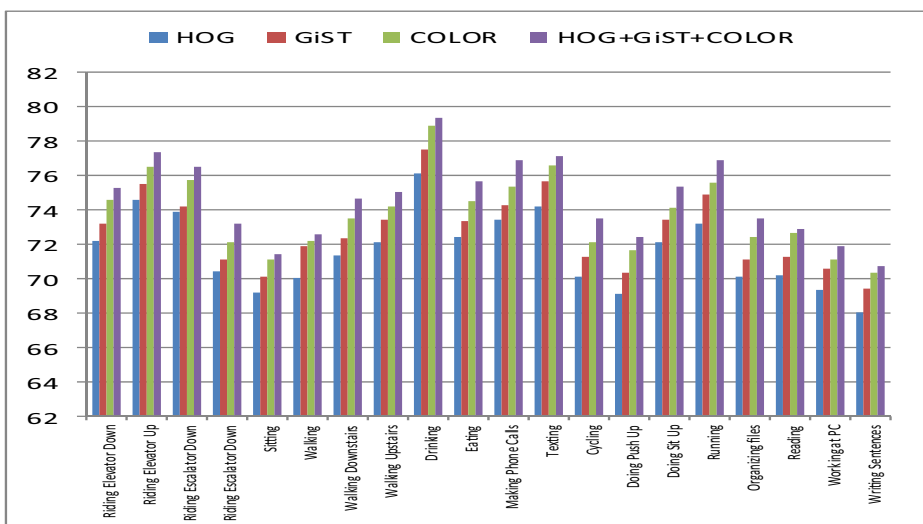


Fig. 16 Second-level activities comparison between different features using random forest

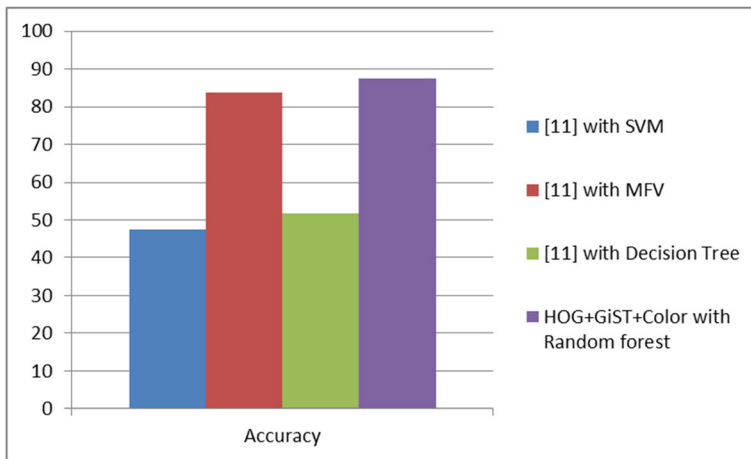


Fig. 17 Comparison of accuracy with existing systems

However, combining HOG, GiST and COLOR feature methods shows that the performance of the Random Forest classifier is better than all the other individual methods (Table 5).

4.7 Evaluation with existing system

The above figure indicates that the proposed system outperforms the other existing systems given in the literature.

5 Conclusion

We gave a broad overview regarding the different problems in the domain of egocentric video that have recently been addressed in the computer vision community. This can be used as a patient monitoring system. We showed that research could roughly be grouped into three categories: object recognition, activity and action detection, and life logging video summarization. We analyzed 1000 samples that included a variety of applications based on activities, which is inadequate in many ways. The activities are categorized into two levels, top- and second-level. With a two-level categorization structure, we can justify the performance gap in activity recognition at two different granularities. Combining the HOG, GiST and COLOR feature methods shows that the performance of the Random Forest classifier is better than all the other individual methods. Random forest provides better results because it handles thousands of inputs; it gives estimates of what variables are important in the classification; it generates an internal unbiased estimate of the generalization error as the forest building

Table 5 Comparison with existing system

| Features | Accuracy |
|-----------------------------------|----------|
| [24] with SVM | 47.5 |
| [24] with MFV | 83.7 |
| [24] with Decision Tree | 51.80 |
| HOG+GiST+Color with Random forest | 87.51 |

progresses; it has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing; and it has methods for balancing error in class population unbalanced datasets. Therefore, it provided better results than the other classifiers.

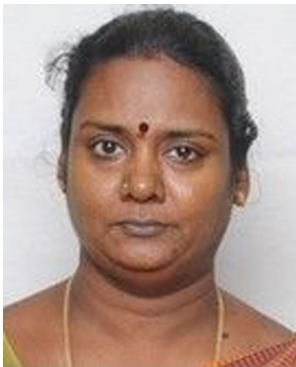
References

1. Bieniek A, Moga A (2000) An efficient watershed algorithm based on connected components. *Pattern Recogn* 33:907–916
2. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection' *Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society Conference*, vol 1, p 886–893
3. de San Roman PP, Benois-Pineau J, Domenger J-P, de Ruyg A, Paclat F, Cataert D (2017) Saliency Driven Object recognition in egocentric videos with deep CNN: toward application in assistance to Neuroprostheses. *Comput Vis Image Underst* 164:82–91
4. Fathi A, Farhadi A, Rehg JM (2011) Understanding egocentric activities. In: *ICCV, Washington, DC, USA*, p 407–414
5. Fong S, Liu K, Cho K (2016) Improvised methods for tackling big data stream mining challenges: case study of human activity recognition. *J Supercomput* 72(10):3927–3959
6. Gemmell J, Bell G, Lueder R (2006) Mylifebits: a personal database for everything. *Commun ACM* 49(1):88–95
7. Hassan M, Ahmad T (2014) A Review on Human Actions Recognition Using Vision Based Techniques. *J Image Graph* 2(1):28–32
8. Hori T, Aizawa K (2003) Context-based video retrieval system for the life-log applications. In: *SIGMM international workshop on multimedia information retrieval*, p 31–38
9. Jacobson L, Kanber B (2015) *Genetic Algorithms in Java Basics*. Apress, New York
10. Knoop S, Vacek S, Dillmann R, Brännström S, Christensen HI (2005) Extraction, evaluation, selection and classification of motion features for human activity recognition. Universität Karlsruhe, Internal Report
11. Kuncheva LI (2014) *Combining Pattern Classifiers: Methods and Algorithms*, 2nd edn. Wiley, Hoboken
12. Li C, Lin M, Yang LT (2014) Integrating the enriched feature with machine learning algorithms for human movement and fall detection. *J Supercomput* 67(3):854–865
13. Maurer U, Smailagic A, Siewiorek D, Deisher M (2006) Activity recognition and monitoring using multiple sensors on different body positions. In: *Proc. Int. Workshop Wearable and Implantable Body Sensor Networks*. <https://doi.org/10.1109/BSN.2006.6>
14. Nibbles JC, Han B, Fei-Fei L (2010) Efficient extraction of human motion volumes by tracking. In: *IEEE Computer Society Conference on Computer Vision, San Francisco, CA, USA*, pp 655–692
15. OpenCV-Introduction to Support Vector Machines http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. Accessed 22 July 2016
16. Ortis A, Farinella GM, D'Amico V, Adesso L, Torrisi G, Battiato S (2017) Organizing egocentric videos of daily living activities. *Pattern Recogn* 72:207–218
17. Rachmadi RF, Ketut Eddy Purnama I Large Scale Scene Classification Using Gist Feature. *Institut Teknologi Sepuluh Nopember Surabaya Indonesia* 60111
18. Pirsavash H, Ramanan D (2012) Detecting activities of daily living in first-person camera views. In: *CVPR, Providence, RI, USA*, p 2847–2854
19. Salman N (2006) Image Segmentation Based on Watershed and Edge Detection Techniques. *Int Arab J Inf Technol* 3(2):104–110
20. Sanal Kumar KP, Bhavani R (2016) Analysis of SVM and kNN Classifiers For Egocentric Activity Recognition. In: *Proceedings of the International Conference on Informatics and Analytics (Pondicherry, India: ACM) August 25–26, 2016*
21. Sanal Kumar KP, Bhavani R (2017) Human activity recognition in egocentric video using PNN, SVM, kNN and SVM+kNN classifiers. *Cluster Computing*
22. Sanal Kumar KP, Bhavani R (2017) Activity Recognition in Egocentric video using SVM, kNN and Combined SVMkNN Classifiers. *IOP Conf Series Mater Sci Eng* 225:012226
23. Solomon C, Breckon T (2010) *Fundamentals of Digital Image Processing*. Wiley, Hoboken, pp 1–18
24. Song S, Chandrasekhar V, Cheung N, Narayan S, Li L, Lim J (2014) Activity Recognition in Egocentric Life-logging Videos, *Computer Vision ACCV 2014 Workshops, Singapore, Nov 2014*
25. Suresha M, Shilpa NA, Soumya B (2012) Apples Grading based on SVM Classifier. In: *National Conference on Advanced Computing and Communications, April 2012*
26. Tukey JW (1974) Nonlinear (Nonsuperposable). *Methods for Smoothing Data. Conference Record EASCON*, pp 673–685

27. Vaca-Castano G, Das S, Sousa JP, Lobo ND, Shah M (2016) Improved scene identification and object detection on egocentric vision of daily activities. *Comp Vis Image Underst*
28. Wang H, Schmid C (2013) Action Recognition with Improved Trajectories ICCV '13 Proceedings of the 2013 I.E. International Conference on Computer Vision, p 3551–3558 December 01–08, 2013
29. Wang X, Gao L, Song J, Zhen X, Sebe N, Shen HT (2017) Deep Appearance and Motion Learning for Egocentric Activity Recognition. *Neurocomputing*
30. Wu J, Wei Z, Chang Y (June 2010) Color and Texture Feature For Content Based Image Retrieval. *Int J Digit Content Technol Appl* 4(3):43–49



Mr. K. P. Sanal Kumar received the M.C.A., degree from Noorul Islam College of Engineering, Kumarakoil, Tamil Nadu in 2000. He received the M.Tech degree in Computer Science and Engineering from Bharath University, Chennai in the year 2005. He has been with Annamalai University, since 2008. He is pursuing his Ph.D in Computer Science and Engineering at Annamalai University. He has published 10 papers in international conferences and journals. He is a life member of CSI and ISTE. His research interest includes Digital Image Processing, Pattern Recognition and Artificial Intelligence.



Dr. R. Bhavani received the B. E degree in Computer Science and Engineering from Regional Engineering College, Trichy in 1989. She received the M.E degree in Computer Science and Engineering from Regional Engineering College, Trichy in the year 1992. She worked at Mookambigai College of Engineering, Keeranur from 1990 to 1994. She has been with Annamalai University, since 1994. She awarded her Ph.D degree in Computer Science and Engineering at Annamalai University, in the year 2007. She published 30 papers in international conferences and journals. Her research interest includes Image processing, Image Segmentation, Image Compression, Image Classification, Stenography, Pattern Classification, Medical Imaging, Content Based Image Retrieval and Software metrics.