



Human Action Detection and Recognition Using SIFT and SVM

Praveen M. Dhulavvagi¹  and Niranjana C. Kundur² 

¹ KLE Technological University, Hubballi, India
praveen.md@bvb.edu

² JSS Academy of Technical Education, Bangalore, India
niranjanckt@gmail.com

Abstract. Human action detection and recognition is the most trending research topic in applications like surveillance of videos, analysis of sports videos and many applications which involve human computer interaction. Many researchers are working on different algorithms to improve the accuracy of human detection. Identifying the actions of human from the given video is a challenging task. In the proposed paper combination of two different techniques is applied i.e. SVM and SIFT techniques are used to identify and recognize the human actions in a given video or image. To extract local features of the given video SIFT based technique is used. In this techniques initially we extract features based on the interest points at a particular point or frames, Mainly SIFT techniques involves 4 basic steps Scale-space extreme detection, Key-point localization, Orientation assignment and Key-point descriptor. Once the key features are extracted they are further classified using SVM classifier. In the results and discussion we perform the comparative analysis of these two techniques on a standard KTH dataset with running and hand clapping actions. The experimental results determine the overall accuracy of 82% for the actions: running and hand clapping actions.

Keywords: SIFT · Scale-space · SVM · Action recognition · Key-points

1 Introduction

Action detection and recognition is active research area for many emerging video surveillance applications in our day today life mainly used for threat identification, monitoring. The primary goal is to identify and recognize the action performed in a given video or image dynamically. Action recognition plays important role in various applications like military video surveillance, video content analysis in sports videos also in many video retrieval and human-computer interaction applications. With the advancement of the technologies action recognition is also used in computer vision domain in recognizing the different human activities such as hand waving, jumping, walking, sitting, standing etc. It can also be used in video content analysis such as scenes with cluttered, moving background, variation in appearance of people, view-points, orientation etc. so to detect and recognize the actions from the above discussed applications is a challenging task so we need to apply appropriate feature extraction and

classification techniques, considering all the above facts has lead into the development of a fully automated human action recognition system for a given video. The system should be capable of classifying a human actions accurately is a challenging task due to problems, such as changes in scale, obstruction, cluttered background, viewpoint and tracking the dynamic motion of a human in the video, so different classes of classification are used based on analyzing the key features. According to the survey actions can be broadly classified into three categories:

- i. Static action for single person: In this case we are considering the actions like standing, sitting, and all the static pose given by the human. Here the input dataset for our application are videos.
- ii. Dynamic action for single person: Here in this case all the actions are dynamic, motion based videos can be considered as input for this class. The actions mainly considered are walking, running, jumping etc. for detection of such actions we require video as an input dataset along with the appearance we need to consider the movement of the person the popular datasets are KTH dataset and Weizmann datasets.
- iii. Multiple people's interactions: This type of actions is relatively complex to recognize, interactions usually include handshaking, hugging, kicking and punching actions. The popular dataset for such actions are UT interaction dataset.

1.1 Action Recognition Process

Action recognition mainly consists of three main stages i.e. Video processing, feature extraction and classification (Figs. 1 and 2).

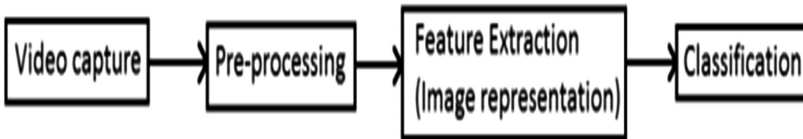


Fig. 1. Stages of action recognition

1.1.1 Video Processing

Prior to video processing a video need to be captured and decoded according to the programming environment, next step is to preprocess the video to eliminate noise and perform foreground and background subtraction.

1.1.2 Feature Extraction

In this stage we extract the key features of the image using standard key feature extraction techniques, In our case we have chosen SIFT (Scale Invariant Feature Transform) for detection and recognition of humans in a given video mainly we need to consider global features because for feature extraction and classification we need global

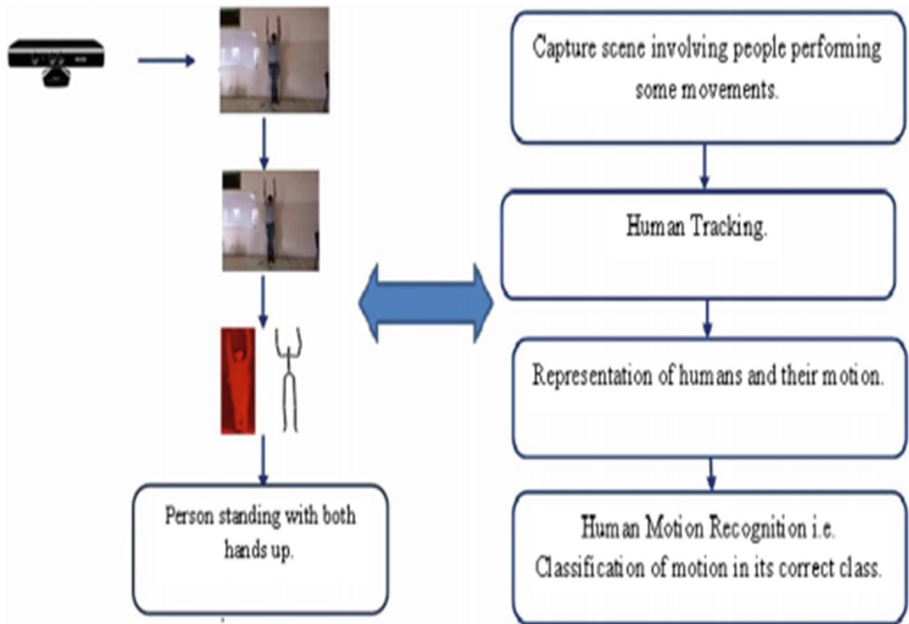


Fig. 2. General framework for human motion recognition

representation of the action. The steps carried out to extract the features are, first we need to localize the human by performing background subtraction, then in second step the ROI (region of interest) need to determine from the images/frames ROI is used to calculate the image descriptor values, since the RoI are more sensitive to viewpoint, occlusion and noise in the given video.

1.1.3 Classification

In this step we classify the actions according to the class labels, there are different classification techniques available among which one to use depends on the feature vectors weather they are of fixed size or varying size along with is it also depends on the properties of the image for example discriminative or generative, probabilistic or statistical classifiers. This means selection of classifier technique depends on the second stage i.e. feature extraction.

2 Related Work

The main focus in action recognition will be on the type of video representations, features with different key formats, different feature extraction techniques are available need to choose the efficient and relevant one, considering all the above facts a survey is carried out, The main aim is to detection of humans in the given video and to extract the

image features and classification of those features mapping to appropriate class labels. In Vision based human action classification a label is assigned for image sequences and action labels, the assignment of labels is a challenging task because of the variation in the inter-personal differences, motion performance, recording settings [2].

2.1 Inter and Intra Class Variations

The performance analysis differs for different actions depending on the variations involved for a particular actions, for example the normal walking action may have different variation like slow walking, fast walking or running this may differ in terms of speed and length, and similar type of observations are there in other action [1]. So an efficient classification algorithm needs to be chosen for classification. For increasing number of actions and class the challenge will be more because similar type of actions may overlap with each other. So considering one single domain and then performing the action classification will be a good alternative according to the authors of paper [3].

2.2 Selection of Human Motion Dataset

According to the literature survey the standard dataset considered for implementation and testing purpose are the standard KTH dataset, This dataset has six actions i.e. boxing, clapping, hand waving, walking, running and jogging. In this proposed paper we are mainly focusing on two actions i.e. hand clapping and running, Here each of these actions involves some variations i.e. walking involves three variations slow walking, fast walking and running so to recognize the particular action we need to consider the viewpoint duration which allows us to recognize a particular action, The background in all these images need to be static [3].

In case of Weizmann human action dataset ten different actions dataset are available but for our experimentation purpose we are considering only two actions walking and hand clapping. in this dataset the background and viewpoint of the image are static for robust evaluation two different sets of action sequences are recorded, we have considered two actions, the Fig. 3 shows the actions considered in which one set shows walking actions picture in different angels and second set shows front-view parallel walking actions [4].

UCF sports action dataset consists of nearly 150 sports actions such as kicking, weightlifting, diving, skating, golf swinging, kicking, running, swinging a baseball bat and walking, there is variation in the performance for different set of classes due to the appearance, background and foreground illuminations along with the view point the Fig. 3 shows the above discussed sample data sets [3, 4].

2.3 Feature Extraction and Representation

Once the appropriate dataset is selected the next phase is to extract the features from the image or video, different feature extraction techniques are available such as SIFT, HoG, NWFE and LKT (Lucas-Kanade-Tomasi). As proposed in reference paper [10]. Extracting the features from the given video or image involves four steps.

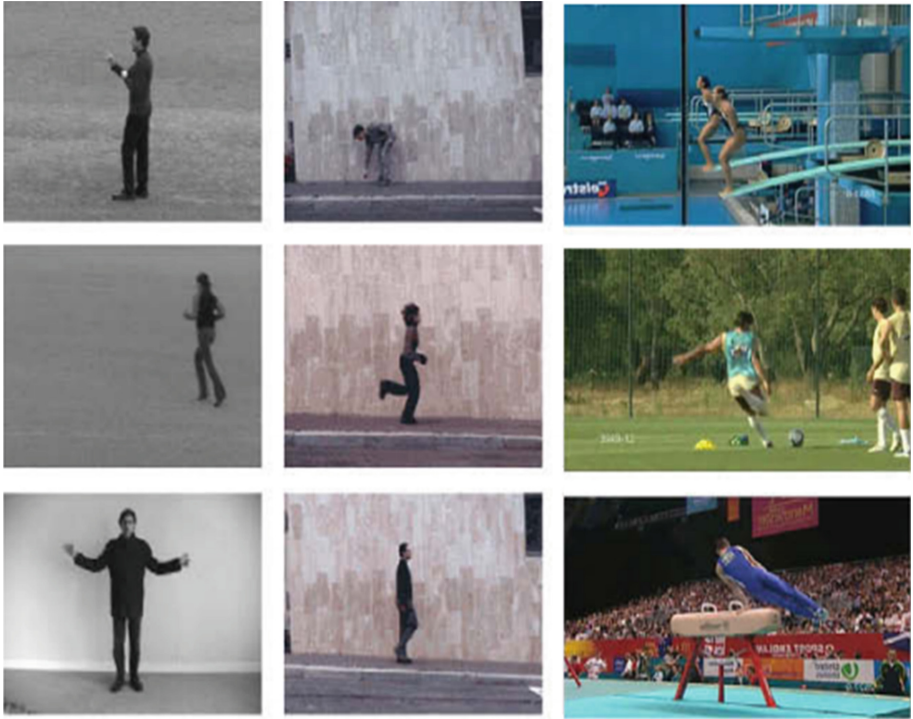


Fig. 3. Examples of KTH, Weizmann, UCF sports actions

2.4 Action Detection and Classification

Once the key features are extracted the next step is detection and classification. To achieve better performance, it is essential to select the relevant and efficient classification algorithm such as SVM, K-NN and HMM. The HMM techniques are used in speech and video applications where time sequential data is involved [11]. The HMM techniques with left-to-right state transition are considered for action recognition. But most of the human actions will be having quasi-period cycles of body movements, so it's complex to model using left-to-right HMMs.

2.5 Support Vector Machine (SVM) Classifier

SVM classifier is most popular classifier, which classifies the given data based on hyperplane representation i.e. on margin-based it is mainly used in pattern recognition applications. The main goal of SVM is to calculate the optimized hyperplane which will maximize the margin of two classes. On this margin the points will be scattered and these data points are called support vectors, Local space-time features can also be

extracted be using SVM classifier which is generally used to recognize the human actions. However the author in [12] has used nonlinear SVM classifier consisting of multi-dimensional Gaussian kernel for action recognition (Fig. 4).

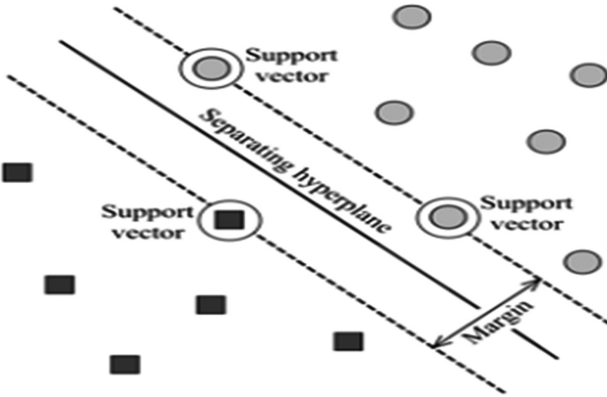


Fig. 4. Linear classification of SVM classifier

K-NN Classifier: K Nearest Neighbour classifier uses a constant value k which is predefined (K) [13], considering this value we try to identify the nearest neighbour and then try to assign label for the data points which fall within this class. In most cases K-NN is used in multi model classification.

3 Implementation

The Fig. 5 shows the overall system model which mainly consists of two phases that interact with the system: Training dataset and test data. Human Action Recognition System consists of three modules: Video pre-processing, Feature extraction and SVM classification. Our training dataset consists of videos of actions running and hand-clapping. Firstly we carry out pre-processing of videos. In the pre-processing step we convert videos into frames. Then these frames are subjected to gray scale conversion as it helps in identifying significant edges or other features and it saves the processing time. Colored images influence the speed of the process. These gray scale images are given as input to SIFT (Scale Invariant Feature Transform) algorithm. The first step of SIFT technique is to extract key features, second step is detection and recognition of human action.

3.1 Feature Extraction

Given an input video the first step is to extract the key features, which basically involves four steps:

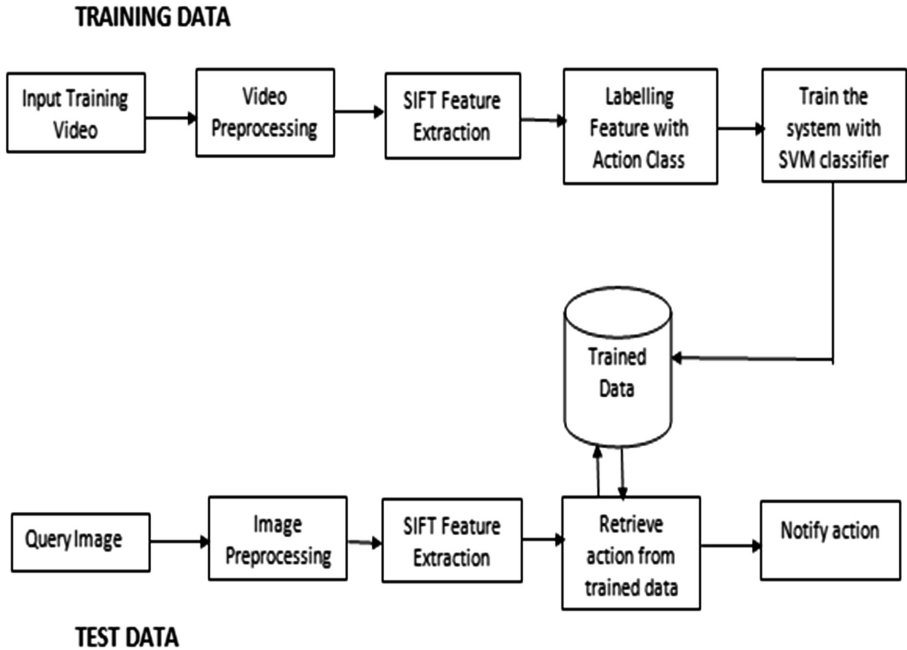


Fig. 5. System architecture

3.1.1 Scale Space Extrema Detection

In this step we create a scale space of images and construct a set of progressively blurred images. Next we consider the difference to get a DoG (Difference of Gaussian) pyramid that occurs at multiple scales. The function to define the scale space

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Here $G(x, y, \sigma)$ is Gaussian variable-scale, $*$ is convolution operator, and $I(x, y)$ is the input image.

Localizing the scale-space extrema, $D(x, y, \sigma)$ is computed by considering the difference between two images, one with scale k times the other. Hence the scale-space extrema function is:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Next step is to identify the local maxima and minima we need to determine local extrema in the scale space and select the key points. For each key point in a $16 * 16$ window we determine the histograms of gradient direction and create a feature vector.

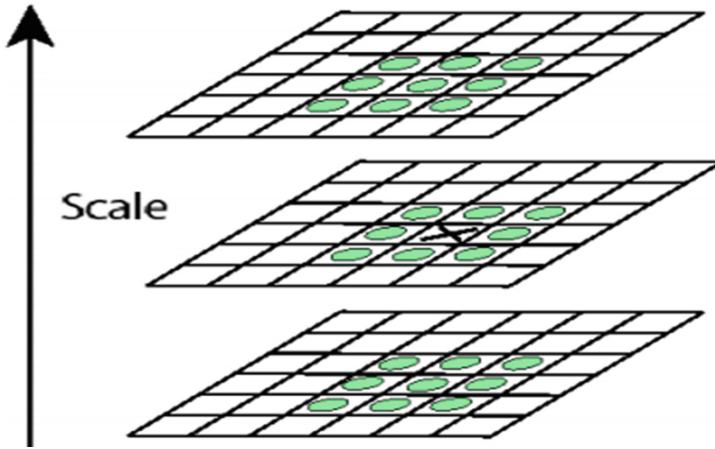


Fig. 6. Keypoint localization

3.1.2 Keypoint Localization

The Scale-space extrema method results in more number of keypoints, all the keypoints are not useful so we need to perform localization and eliminate some of the keypoints which are unstable, sensitive to noise and have low contrast such points will be eliminated. In the Fig. 6 we can observe that each point is compared with its 8 neighborhood points of the current image, and 9 neighbors each in the scales above and below. Generally in this method we try to find the robust extremum (maximum/minimum keypoints both in space and in scale). Basically we need to use DoG pyramid to determine maximum values and then using any of the edge detection technique we need to eliminate “edges” and pick only corners.

The function for determining the scale $f = \text{kernel} * \text{image}$

Kernels are:

Laplacian

$$L = \sigma^2(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

Difference of Gaussians (DoG):

$$\text{DoG} = G(x, y, k\sigma) - G(x, y, \sigma)$$

In Laplacian method location of extremum, z is given by the following equation:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

If the z value falls below the threshold value then that such point will be discarded and rest of the points are considered for feature extraction. The Fig. 7 shows the sample image for determining the extrema at different scale.

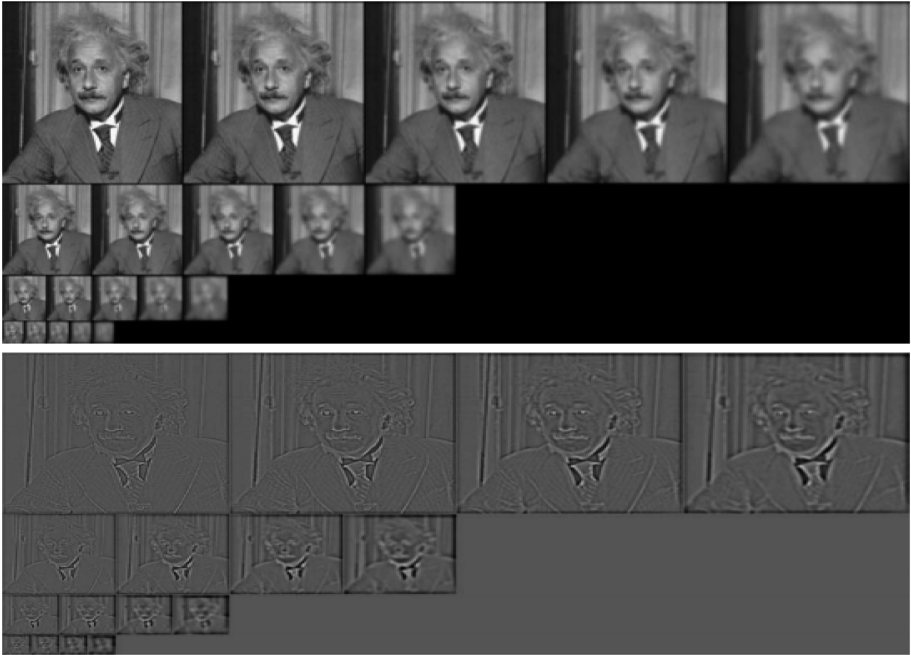


Fig. 7. Finding extrema at different scale

3.1.3 Orientation Assignment

Construct the histogram of local gradient directions at selected scale and assign each keypoint with one or more orientations based on local image gradient directions. Each keypoint now specifies stable 2D coordinates $(x, y, \text{scale}, \text{orientation})$. The keypoints which are obtained from localization are assigned with consistent orientation considering the image properties, Orientation can be calculated considering the keypoint scale technique for the selection of the Gaussian smoothed image value L , next we need to compute the gradient magnitude m using the function:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

Next we compute the orientation $\theta(x, y)$ considering the pixel differences:

$$\theta(x, y) = \arctan 2(L(x, y+1) - L(x, y-1), L(x+1, y) - L(x-1, y))$$

3.1.4 Keypoint Descriptor

Once the keypoints locations at particular scales and assigned orientations are determined, next we need to compute descriptor vector for the local image regions and keypoints which are invariant with respect to the viewpoint and illusion. The gradient

value obtained using orientation assignment is used to construct the keypoint descriptor. Once the descriptor values is calculated next step is to rotate the gradient value upwards upto the orientation of keypoints and then weighted by a Gaussian with variance of $1.5 * \text{keypoint scale}$.

The Fig. 8 shows the 484 gradient window with histogram of 484 samples per window in 8 different directions, here in this case the Gaussian weightage around centre (σ) value is 0.5 times that of the scale keypoint, so totally it is $4 * 4 * 8 = 128$ dimensional feature vector. The Fig. 9 shows the sample example of SIFT technique processing pipeline which is discussed in above section. The extracted features are labeled with action class and then the system is trained with SVM classifier. The SVM creates a hyper plane for classifying the data into a high dimensional space for separating data with different labels. Using training dataset a model is built. Now the test data is processed with the same steps as that of training dataset but for predicting the action. These test images are those that are not trained earlier. Finally both the files consisting of labeled features with action classes of training dataset and test data are given as input to SVM classifier. The SVM classifier predicts the action based on the similarity measure between training data and test data and then detects the action.

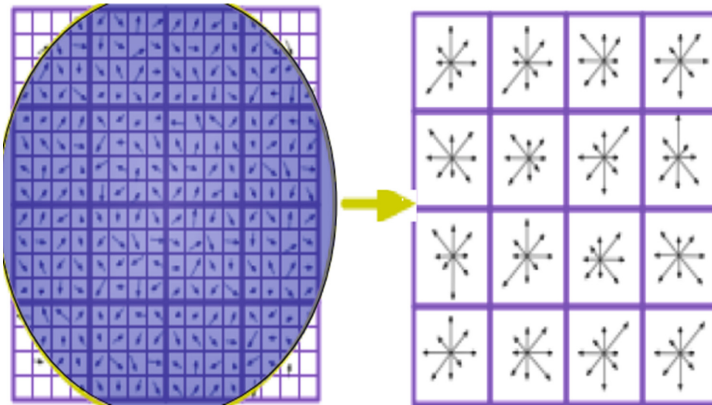


Fig. 8. Image gradient and keypoint descriptor

3.2 Video Pre-processing

Input for video preprocessing module is KTH video dataset and output is gray scale images. This module converts the input video to frames and perform image enhancement (i.e. Noise removal etc.). Based on standard convention, 26 frames are generated per second. So the number of frames generated per video depends on the length of the video. Certain frames that are generated are not informative i.e. the frames that doesn't contain complete human postures are to be eliminated. Hence these key-frames are converted to gray scale images (Fig. 10).

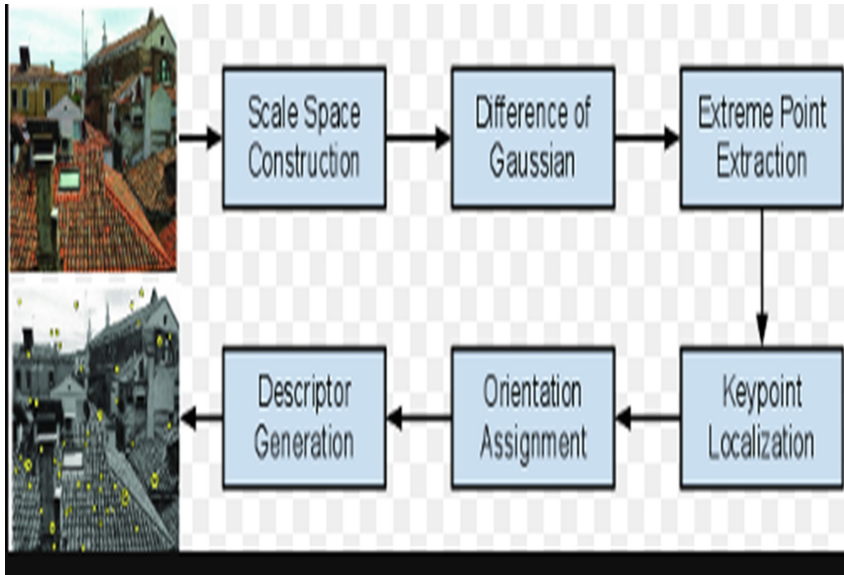


Fig. 9. SIFT algorithm processing steps

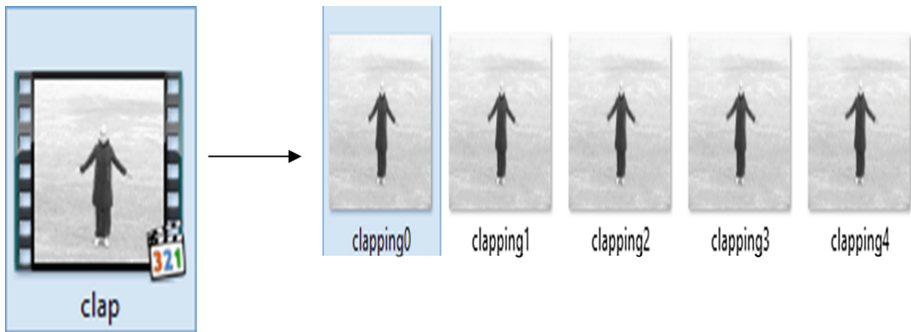


Fig. 10. Video to frames extraction

The input for SIFT technique is a gray scale images and output is local features of the image. For the given input image keypoints are determined considering the scale-space extrema, the low contrast points and edge response points along an edge are rejected. SIFT performs the following steps for extracting local features: Represent the image as the DoG pyramid, to create this pyramid the image should be scaled to 4 different sizes also called octaves. For each octave a set of images is created each with different degree of Gaussian blurred, each image is extracted with one degree less blurred, after doing this for each octave consolidate all octave and put them together to form difference of Gaussian pyramid (DoG). The Fig. 11 shows the process to determine the difference of Gaussian pyramid.

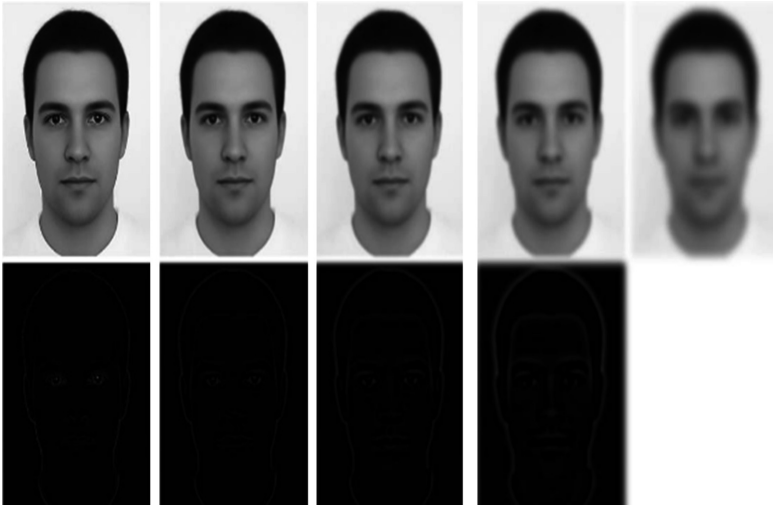


Fig. 11. Scale space extrema detection process

Next step is keypoint localization where we identify the keypoints in scale space representation this will have too many points so we need to discard flat areas and edges keeping corners. Next discard unstable extrema and keep extrema that are found at several scale space planes, at this point we should have good set of key points which will be used for orientation assignment. In orientation assignment step the keypoints are oriented based on histogram of gradients, this is done to achieve rotation invariance, at each pixel we try to find derivatives and that will give us the magnitude and direction.

Considering the interest of points and looking at the orientation around each of the points along all the directions and next we try to build the histogram of 36 bins. We construct the orientation histogram consisting of orientation histogram with 36 bins covering 360° is built. Next it is weighted by gradient magnitude and gaussian-weighted circular window with equal to 1.5 times the scale of keypoint. Assign weights for each of the pixels based on the gradient magnitude next step we transform the data to its relative to the assigned orientation, scale and location providing invariance to these transformations.

3.3 Key-Point Descriptor

At this step we will consider location, orientation and scale as keypoints. Consider the neighborhood around the future points and take the gradient directions of other points and quantize them into 8 different directions to form a histogram the Fig. 12 shows the $16 * 16$ neighborhood around the point and $4 * 4$ histogram array with 8 orientations forming 128 dimensions (Fig. 13).

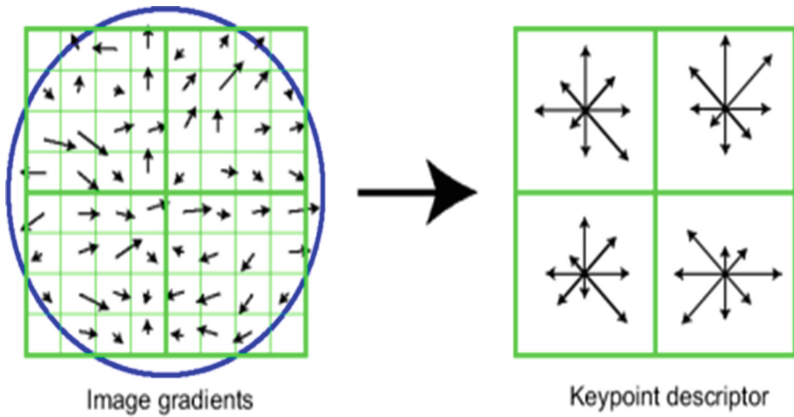


Fig. 12. Keypoint descriptor extraction process

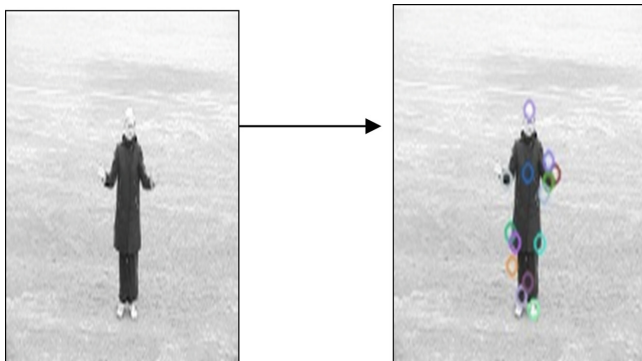


Fig. 13. SIFT keypoints extraction

3.4 SVM Classifier

A two-layer SVM classifier is used at the first level the keypoint descriptors derived from SIFT technique are used for classification, A key-point features, labels are given as input for the classifier. The goal of SVM (Support Vector Machine) is to maximize the margin of training data by finding the optimal separating hyper plane. Among all hyper planes, SVM selects the hyperplane as optimal hyperplane where the distance of hyper plane is as large as possible. Generally SVM is applied on two types of data: Separable data and non-separable data [8].

4 Results and Discussions

See Fig. 14.





Fig. 14. Computed SIFT key points

4.1 Confusion Matrix

We have drawn a confusion matrix for running and hand clapping actions for KTH and Weizmann dataset. We have trained 3 videos as a sample dataset and we have considered 100 images of KTH and Weizmann for each action as our test data. Out of 100 running images of KTH dataset, 74 were recognized correctly as running and 27 images were not recognized. Out of 100 running images of Weizmann dataset, 62 were correctly recognized as running and 38 images were not recognized. Out of 100 hand clapping images of KTH dataset 89 were correctly recognized as hand clapping and 11 images were not recognized. Out of 100 hand clapping images of Weizmann dataset 78 were correctly recognized as hand clapping and 22 images were not recognized. The system is 81% efficient.

Table 1. Test case results

Datasets	Classes	Videos	Efficiency
KTH 	<ul style="list-style-type: none"> • Number of action classes = 2 • Verbs: Running and Hand Clapping. 	<ul style="list-style-type: none"> • 50 videos (10 training, 4 testing). • Resolution = 160x120. • Black and white videos. • Static camera 	<ul style="list-style-type: none"> • 73% for running and • 89% for Hand clapping
WEIZMANN 	<ul style="list-style-type: none"> • Number of action classes = 2 • Verbs: Running and Hand Clapping. 	<ul style="list-style-type: none"> • 20 videos (evaluate by leave one out cross validation). • Resolution = 180x144. • Static camera 	<ul style="list-style-type: none"> • 62% for Running and • 78% for Hand clapping

The Table 1 shows the dataset considered for testing purpose the number of action classes initially considered were 2 i.e. hand clapping and running, in the Table 4 we had considered 5 different action classes were considered and tested, for KTH dataset we had considered 50 videos which were having a resolution of 160×120 which were captured from static camera and these 50 videos were processed, the efficiency of the system was 73% for running action and 89% for hand clapping, similarly for Weizmann dataset we had considered 20 sample videos with resolution of 180×144 , the efficiency of the system was 67% for running and 78% for hand clapping.

Table 2. Confusion matrix for KTH data set

KTH dataset	Actions	
	Running	Hand clapping
Running	0.74	0.27
Hand clapping	0.11	0.89

Table 3. Confusion matrix for Weizmann data set

Weizmann dataset	Actions	
	Running	Hand clapping
Running	0.62	0.38
Hand clapping	0.22	0.79

The Tables 2 and 3 shows the confusion matrix for KTH & Weizmann dataset, here we have considered two actions as test cases running and hand clapping actions for KTH and Weizmann dataset. We have trained 3 videos for each action and given to SVM classifier. We have considered 100 images of KTH and Weizmann for each action as our test data. Out of 100 running images of KTH dataset, 73 were recognized correctly as running and 27 images were not recognized. Out of 100 running images of Weizmann dataset, 62 were correctly recognized as running and 38 images were not recognized. Out of 100 hand clapping images of KTH dataset 89 were correctly recognized as hand clapping and 11 images were not recognized. Out of 100 hand clapping images of Weizmann dataset 78 were correctly recognized as hand clapping and 22 images were not recognized. The system is 81% efficient.

The performance analysis of proposed approach and the related works are all done considering both KTH and Weizmann dataset which is discussed in Tables 4 and 5.

Table 4. Confusion matrix using our approach

	Running	Hand clapping	Hand waving	Walking	Jogging
Running	0.90	0.08	0.0	0.0	0.0
Hand clapping	0.06	0.88	0.0	0.0	0.2
Hand waving	0.0	0.0	0.86	0.3	0.0
Walking	0.0	0.0	0.13	0.83	0.0
Jogging	0.7	0.0	0.0	0.0	0.86

Table 5. Performance comparison with other methods

Method	KTH	Weizmann
Proposed method	97.89	96.66
Bregonzio et al.	94.33	95.66
Tran et al.	95.66	0.00
Kaaniche and Bremond	94.6	0.00

5 Conclusion

Action recognition in computer vision and video surveillance applications has increasing demand, in this paper combination of SIFT and SVM techniques are used for feature extraction and action recognition, the dataset considered for testing purpose are KTH and sports, Weizmann dataset there are totally 12 actions in KTH dataset, in the proposed paper we have tested the application considering two actions running and hand clapping, on this input dataset SIFT algorithm is applied to extract the feature vectors which will be further used to classify the action using SVM classifier. The proposed method is very robust considering the different variations in human actions. In future we can consider other complex actions and activities involving multiple people and cluttered background for recognizing actions. We can apply more robust algorithms for feature extraction and classification.

References

1. Poppe, R.: A survey on vision-based human action recognition. *J. Image Vis. Comput.* **28**(6), 976–990 (2009). Human Media Interaction Group, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente
2. Kellokumpu, V., Zhao, G., Pietikäinen, M.: Recognition of human actions using texture descriptors. *Mach. Vis. Appl.* **22**(5), 767–780 (2009). <https://doi.org/10.1007/s00138-009-0233-8>
3. Bermejo Nievas, E., Deniz Suarez, O., Bueno García, G., Sukthankar, R.: Violence detection in video using computer vision techniques. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011. LNCS, vol. 6855, pp. 332–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23678-5_39
4. Schuldts, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR Proceedings of the 17th International Conference, vol. 3 (2004). <https://doi.org/10.1109/icpr.2004.1334462>
5. Ahmad, M., Lee, S.W.: Human action recognition using shape and CLG-motion flow from multi-view image sequences. *Pattern Recogn.* **41**(7), 2237–2252 (2008)
6. Patil, R.A., Sahula, V., Mandal, A.S.: Facial expression recognition in image sequences using active shape model and SVM. In: Fifth UKSim European Symposium on Computer Modeling and Simulation (EMS), pp. 168–173 (2011)
7. Lai, K.-T., Hsieh, C.-H., Lai, M.-F., Chen, M.-S.: Human action recognition using key points displacement. In: Elmoataz, A., Lezoray, O., Nouboud, F., Mammas, D., Meunier, J. (eds.) ICISP 2010. LNCS, vol. 6134, pp. 439–447. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13681-8_51
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110 (2004)
9. Hoang, L.U.T., Ke, S., Hwang, J., Tuan, P.V., Chau, T.N.: Quasi-periodic action recognition from monocular videos via 3D human models and cyclic HMMs. In: Proceedings of IEEE International Conference on Advanced Technologies for Communications (ATC), Hanoi, Vietnam, pp. 110–113 (2012)
10. Scheldts, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR), Cambridge, UK, vol. 3, pp. 32–36 (2004)
11. Kumari, S., Mitra, S.K.: Human action recognition using DFT. In: Proceedings of the Third IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Hubli, India, pp. 239–242 (2011)
12. Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., Huang, T.: Action detection in complex scenes with spatial and temporal ambiguities. In: Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV), pp. 128–135 (2009)
13. Bregonzio, M., Xiang, T., Gong, S.: Fusing appearance and distribution information of interest points for action recognition. *Pattern Recogn.* **45**(3), 1220–1234 (2012)