Chapter 1

Introduction

1.0: Overview of HAR

In recent years, human action recognition (HAR) has evoked great interest in computer vision due to its various applications and need in Surveillance, cognitive assistance, indoor localization and tracking, fitness monitoring, biometrics, health care, Ambient Intelligence, and Human–Computer Interaction systems [1] [2]. Since HAR system has become a recent field of research, human action classification has come out with much advancement [3] [4]. The changing shape and size of a person moving in different directions with respect to a single camera affects HAR system. The other challenging factors include illumination condition, body postures variations, occlusion, cluttered background and performance rate etc [5]. The system which can adapt to these changes and perform classification accurately is considered as a sober HAR system. The task is challenging due to the environmental conditions and human body taxonomy.

There are challenges apart from these above cited challenges which include advancing the system such that there can be a generalised technique to recognize any activity. Action localisation in spatial and temporal domain in video segments and gathering enough number of training samples such that an activity can be recognised accurately. The aim of this work is to recognise and classify the actions according to their uniqueness and improved methodology such that action recognition can be done with high accuracy and faster execution.

As the name suggests, “action recognition” means recognizing the human action by the system that analyses the video sequences and extract the features from the sequences to learn the system for different action classes and then uses the learned knowledge to classify the new action among those learned classes. Nowadays, researchers are highly motivated towards improving HAR systems due to the increasing popularity and great prospective of activity recognition . They made HAR systems to work under various challenging environment, real time noisy data, to work for multiple users, escalating security and privacy and minimizing the amount of training data to reduce the memory required by system. Smart phones are introducing new development in this area. Smart phones instigate new issues such as draining of battery and complexity in computation.

This thesis mainly focuses on vision based human activity recognition that acquires the video from camera as a primary sensor and determine the activity in a video segment by analyzing the video. In this work, a novel approach is proposed in the feature extraction part of recognition system for accurate activity recognition like running, jogging, skipping, boxing, etc. There are two types of feature descriptors, namely shape and motion based descriptors [6]. Silhouette of the human body is the outline and dense shape which contains the contour of the human shape when seen from the visible light against a brighter background. Silhouette of human body is represented for shape based descriptor. Motion based descriptors are rooted on the movement of the body, and the area of interest can be exploited using frame differencing, optical flow, background modelling. Motion based descriptors are inefficient when the object of interest is moving with continuously changing speed in subsequent frames [7]. This task is crucial and full of challenges.

The philanthropy given by this work for efficient representation of human actions is in feature extraction. Firstly, a entropy based texture segmentation is used for foreground detection in which human activity silhouettes are extracted from the video sequences. The various silhouettes from the activity are used to get the average energy image (AEI) features for each activity. The average energy image is a unique feature for an activity in such a way that the actions performed by the same person i.e. intra-class variations are reduced and the actions performed by different persons are maximized, such that more discriminative information can be extracted for classification.

The spatial distribution gradient descriptor is applied on average energy images for computation of feature vector. The SDG is computed for three levels and instead of going to the fourth level, the previously computed three level parameters are concatenated for reducing the dimension of feature vector. SDG have a disadvantage of finding features in single orientation. To overcome this problem, spatio temporal keypoints are applied to the different frames of the videos for effective representation of human activity. They extract information by changing scale and orientation. This hybrid feature is exploited for each human activity category of dataset, on which support vector machine is applied for classifying the activity performed in a video. The same feature vectors are used to train hidden markov model and the accuracy of HMM is compared with that of SVM.

DATABESES:

KTH dataset

Schuldt et al. introduced the KTH dataset in 2004 and has various challenges in comparison to the Weizmann dataset [8]. This dataset comprises of six activities, namely; ‘‘Hand-Clapping,’’ ‘‘Hand-Waving,’’ ‘‘Jogging,’’ ‘‘Jumping,’’ ‘‘Running,’’ and ‘‘Walking’’. There are 100 videos in each activity in different conditions. These sequences are recorded with a static camera in uniform background at a frame rate of 25 frames per second and having a spatial resolution of 160×120. There is a significant movement in camera while recording due to which segmentation is a challenging task. Texture based segmentation is used in this context.

Weizmann dataset

Gorelick et al. introduced this dataset in 2007 [9]. This dataset comprises of 10 activities, namely run, walk, jump, gallop-sideways, jack, pjump, wave1, wave2, skip, bend. Each video sequence has a frame rate of 25 frames per second at a spatial resolution of 144×180. There are a total of 9 videos in each activity performed by 9 different actors with a total of 90 video sequences. These sequences are recorded with a static camera. This dataset is less challenging as compared to KTH dataset.

Ballet Dataset

The Ballet Movement activity information set (Fathi, 2008) is one of the complex human activity information sets [10]. This information set comprise of eight Ballet Movements performed by three on-screen characters and these movements are named as Hopping , Jump, Left-to-Right Hand Opening, Leg Swinging, Right-to-Left Hand Opening, Standing with Hand Opening, Stand Still and Turning. The information set is exceptionally challenging because of the extensive measure of intra-class dissimilarities in terms of spatial and temporal scale, speed, and attire.

IXMAS Dataset

With the reason for developing the experimentation of our technique to a more troublesome dataset with more camera perspectives, we have picked the IXMAS dataset which is mainstream among human activity recognition techniques that are particularly intended for multi view recognition. The INRIA Xmas Motion Acquisition Sequences (IXMAS) dataset (Weinland et al., 2006) incorporates multi-view information furthermore, is particularly gone for perspective invariance testing [11] . It gives 390×291 px resolution images from five diverse points including four sides and one top-view camera. An arrangement of 12 on-screen characters have been recorded performing 14 distinctive activities (check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, throw over head and throw from bottom up) 3 times each, subsequent in a dataset with more than 2000 arrangements. This benchmark exhibits an expanded trouble since subjects were asked to openly pick their position and introduction. Along these lines, every camera has caught distinctive review edges, which makes strategies which depend on altered camera sees (front, side, and so forth.) inadmissible.

1.1 Motivation for Human Activity Recognition

In this chapter, we tend to first give the motivation for activity Recognition. Then the challenges and different steps in Action Recognition with flow-chart are mentioned. Then we provide the activity classification approaches in step according to our taxonomy.

Human action recognition is a very vital component of video surveillance systems for event based analysis of surveillance videos. Video based surveillance systems play a very important role in the cases where regular patrolling by human beings is not possible like international border patrolling, nuclear reactors etc. Demand for automatic surveillance systems in civilian applications like monitoring a parking lot, shopping complexes etc. is also increasing heavily. It is difficult and manpower intensive to monitor the data collected from various cameras continuously and this gives rise to the necessity for automatic understanding of human actions and building a higher level knowledge of the events occurring in the scene by the computer vision system.

Analysis in surveillance scenarios often requires the detection of abnormal human actions. Most of the normal human activities are periodic like walking, running etc. Lack of periodicity is therefore an important cue of an activity being deviant from the normal. Consider for example a typical event of surveillance interest: exchange of brief cases by two agents. The scene essentially consists of an agent walking across the scene who then bends to lift up or leave the briefcase. This event can be described as concatenation of walk-bend-walk actions, where bend is deviant from normal behaviour. However abnormal events and therefore abnormal human activities are context dependent and may vary for different situations. For example, in a shopping mall where people normally walk from one counter to another, running could be defined as an abnormal action and could be an event of interest for surveillance purposes. This calls for a need of unified framework for detecting and recognizing both periodic and non periodic human actions.

Recognition of human movements has also been exploited to a large extent for animation like avatar control, for giving gesture based commands to virtual reality interfaces, human computer interactions in smart room like environments etc. Content based video retrieval, indexing and searching is also becoming popular these days with the concepts like Video Google coming up. These systems require cognitive vision techniques for analyzing videos which in real life scenarios mostly converges to analyzing human actions in the videos. Video annotation of sports videos is an excellent example of this category where complex human sport actions are required to be classified. A good discussion on the promising application scenarios and the suitable approaches in these scenarios can be obtained. The wide scale applications of human activity analysis and various challenges involved at different stages in building this system makes it a demanding area of research in computer vision.

1.2 Framework of HAR

In this section, we describe the framework of a typical human activity recognition system. Figure 1.1 shows the various blocks involved in any HAR system. Each block is an area of recent research in its own way. This work mainly focuses on the feature extraction part of HAR system.

Human body has countless of flexibility. Demonstrating structural and dynamic features for activity recognition of such a mind boggling object is an intense undertaking. Analyzing human action is a challenging task due to the non rigid and self occluding characteristic of the verbalized human movement.

Executing genuine action recognition framework is an overwhelming errand considering the difficulties at every phase of the framework like background disorder, dynamic enlightenment changes, and camera motion and so on in background subtraction stage, partial impediments in the tracking and feature extraction stages. The execution of the recognition stage relies on upon these past stages furthermore on the decision of features for activity representation. The activity classification issue is characterized by huge intra class variability presented by different sources like the adjustments in camera perspective, anthropometry (body shapes and sizes of various performing artists), distinctive dressing styles, changes in execution rate of movement, individual styles of on-screen characters and so forth [1].

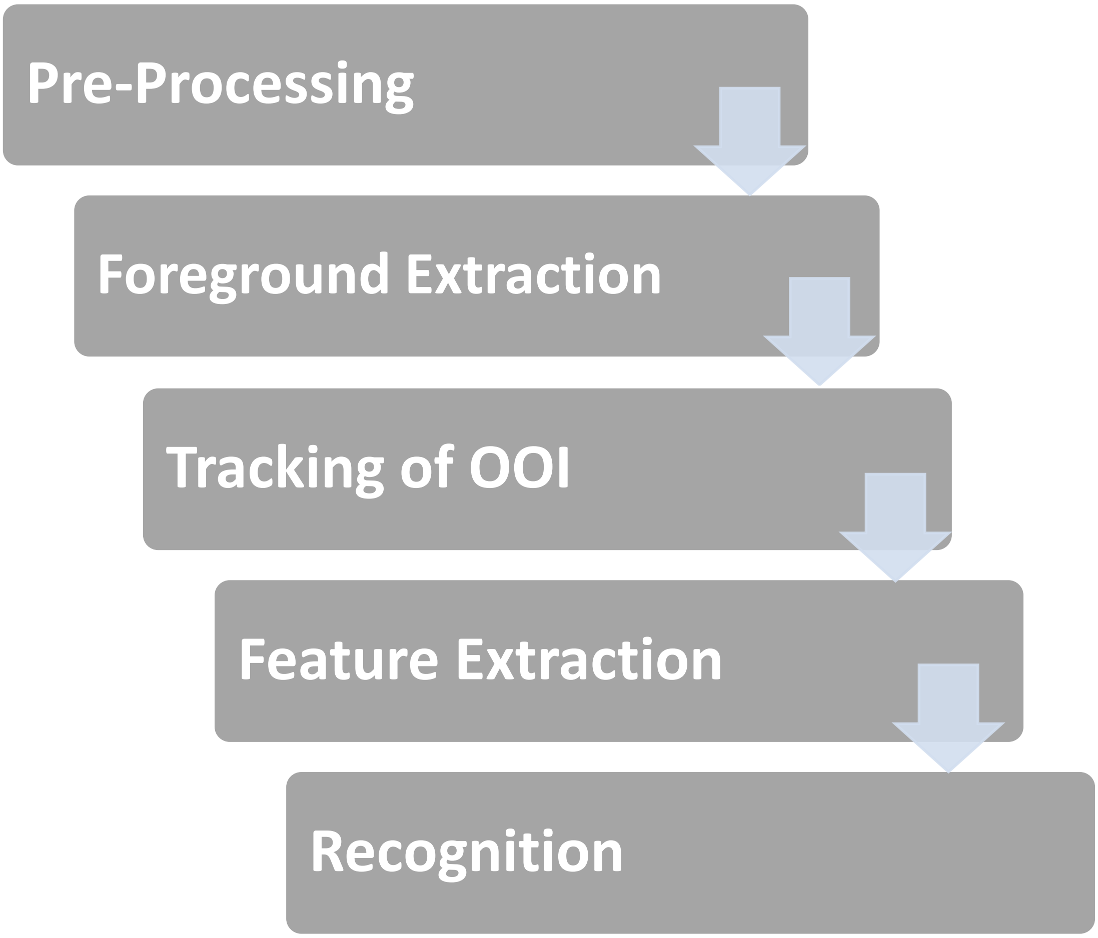


Figure 1.1: general Human Action Recognition Framework

To start with the framework is initialized. Next, the subject is then followed to acquire the position in the scene, which infers a figure-ground division to isolate the subject from the foundation. Posture estimation may be a piece of the yield or be utilized as info to the recognition procedure. Recognition groups the activities performed by the subject. A framework does not have to incorporate every one of the four procedures, and some procedures may be certainly performed by one of the consequent procedures. Case of this can be found in real life recognition frameworks depicted in the accompanying segments, where activities are arranged in a clasp, yet the position and posture of the performer is obscure.

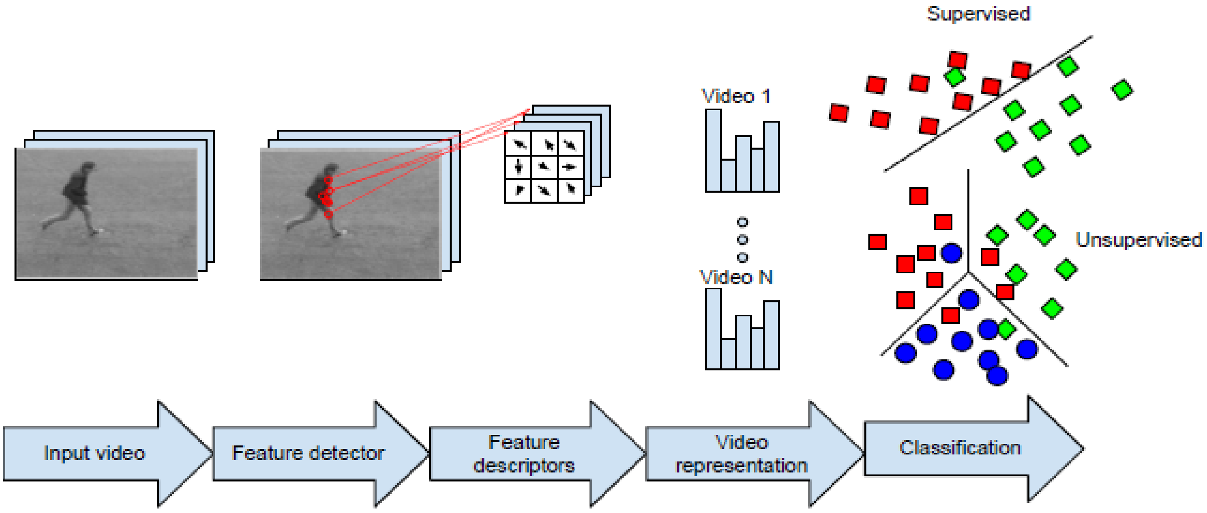


Figure 1.2: HAR framework used in our approach

The structure comprises of four primary segments: feature detector, feature descriptor, video representation and classification. The distinctive parts of this system can be replaced with diverse algorithms to shape distinctive mixes and in this way deliver distinctive results. Figure 1.2 depicts the steps in HAR system.

The initial step is to detect interest points in the video, which are the positions where the feature descriptors are processed. These points should situate at spots in the video where the movement is taking place. The feature descriptor encodes the data in the zone of an interest point into a representation suited for representing the activity. The feature descriptor ought to be invariant to changes like orientation, scale and illumination to be able to match features across different kinds of videos.

The arrangement of local feature descriptors in a video must be consolidated into a representation that empowers the correlation with other videos. The most prevalent technique is spatio temporal interest key points, where the spatial and temporal locations of the features are considered. Different strategies try to take the relationship between the features into record. The classification step can be: unsupervised, semi-supervised or supervised. In the unsupervised approach, we assume that we do not know the labels of any of the videos. The videos are then grouped together based on their similarities. The number of groups used for unsupervised learning can be given as part of the problem, or can be dynamic where different partitions of videos corresponds to different semantically meanings. In semi-supervised classification there is some prior knowledge about the videos, which can consist of a few labelled samples, or e. g. a constraint saying that sample a and b are from different classes without giving the label. Semi-supervised learning is not explored further in this thesis. Supervised classification uses a large number of training samples to train a classifier.

1.3 Challenges in HAR

In this section, we tend to list out a number of the challenges researchers faced in action recognition and describe the various ways employed by researchers to handle them. A quantitative performance comparison of the proposed techniques is troublesome since datasets and testing strategy used vary considerably. Nonetheless, the number of training information and ability to generalize the strategy to any variety of action will be used as benchmarks to classify the methods as these measures are crucial for real-world problems.

1.3.1 Variation in Viewpoint

Most ways assume that action is performed from a set viewpoint. Fig.1 shows why researchers tend to form this assumption. The figure 1.3 shows four views of an actor acting walking action. In every point of view, the location and posture of the person vary significantly [12] [13]. Additionally, motion patterns in each view would seem totally different, creating recognition of the action not so trivial.

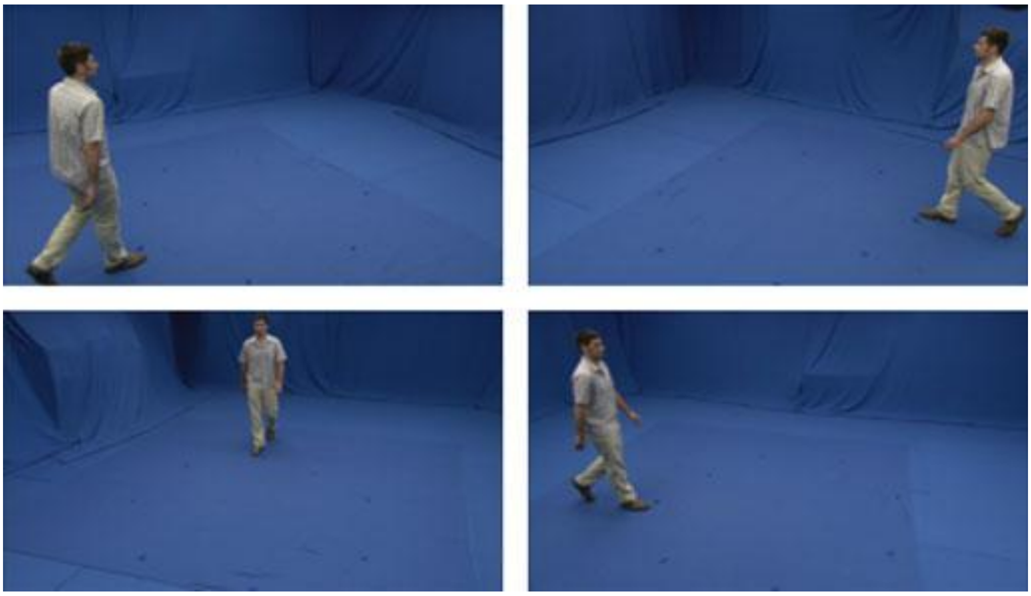


Figure 1.3: Walking action images from i3DPost multiview dataset.

The solution to handle the variation in viewpoint problem is to train the classifier using multiple camera views. This is done by quantizing the viewing angle into different segments of angle. For each segment, features are extracted and are combined to train a single classifier so that the machine can handle any variation in viewpoint or used to train a set of classifiers, one for each angle viewpoint to work for multiple camera views.

1.3.2 Occlusion

The existing systems need that the action being performed should be clearly visible within the video sequences [14]. During a traditional surveillance video, this is often unfeasible as a result of the number of people within the field of view of the camera. Occlusions will be either self occlusions or those created by alternative objects in the field of view of the camera throughout the video capture. This poses an enormous challenge to the research community as not all the body parts playing the action are visible within the video sequence [15].

One set of researches are going to explore classifiers that may handle occlusions. Since feature extraction from occluded components isn't attainable, it's necessary to come up with robust classifiers that may adapt according to the presence of occlusions [16].

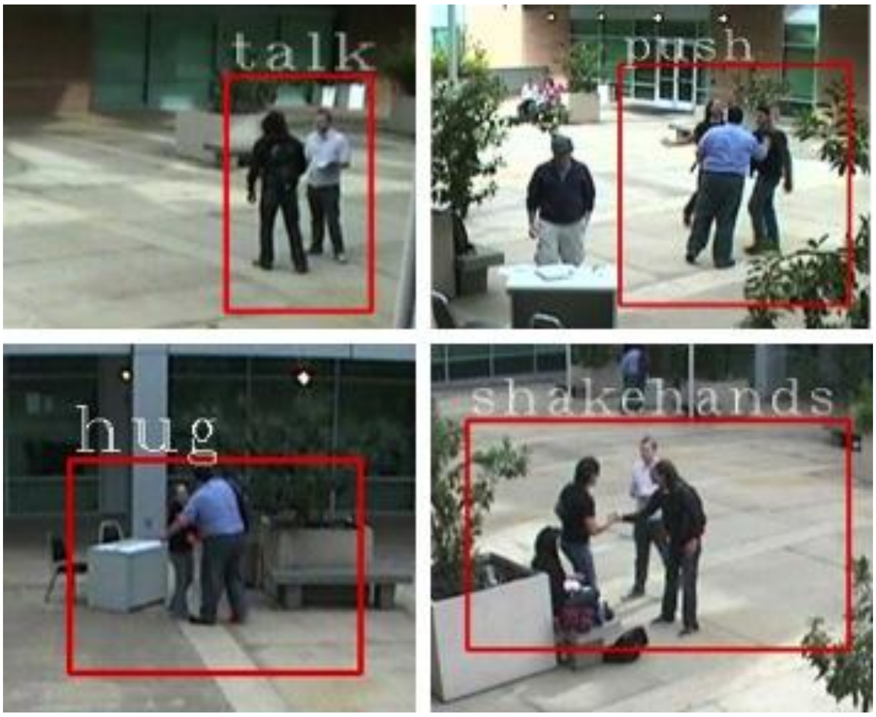


Figure 1.4: Actions during occlusion.

In figure 1.4, four actions are being performed which are depicting occlusion problem. Image showing push action is under occlusion as the person hand's that is pushing the other two persons are not clearly visible. In hug action, one person is behind the other person, so the action is not clearly visible. When occlusion occurs, it is hard to find the feature vector appropriately hence the classifier should be robust enough to classify the images which occlusion problem.

1.3.3 Execution Rate

Each individual performs associate action at his/her own pace. In addition, there's no guarantee that someone can repeat the action at a similar speed each time. This variation within the rate of execution of an action should be taken into consideration in an action recognition system.

The methods providing probabilistic framework is suitable for handling these type of challenges. Hence Hidden Markov Model (HMM), fuzzy systems, conditional random fields, Bayesian networks etc are better suited classifiers [17].

1.3.4 Variation in Body measures

Each person has completely different body size, proportion, and comfort zone while performing an action [18]. As an example, a waving gesture of someone may involve moving the hand higher than the pinnacle and then wave the hand; however another person won't move his hand above his head and would simply wave from a shoulder height. Thus, researchers develop a generalized approach to capture and handle these variations.

1.3.5 Camera Motion

Mostly in human action recognition systems, researchers assume static cameras for acquisition, which could not be the case in unconstrained systems. HAR system is severely affected by camera motion since features are extracted from erroneous and misleading motion patterns which are induced in the videos [19]. Shape based feature extraction generally require a good background model, stationary cameras and better tracking mechanism. Background subtraction required by these features is littered by moving cameras.

1.3.6 Cluttered background

Dynamic or cluttered background is a form of distraction in the video sequence from the original action of interest as it introduces ambiguous information. Flow-based methods that calculate motion are affected as they detect unwanted background motion along with the actual required motion. In addition, color-based and region-based segmentation approaches require uniform non varying background for reliable segmentation and tracking of the foreground object [20]. To avoid the anomaly introduced, most applications assume a static background or a method to handle background segmentation from the videos prior to processing

1.4 Outline of thesis

The thesis is organized in the following way:

1. Chapter 1: **Introduction** introduces the context and motivation of the research presented in this thesis.
2. Chapter 2: **Related work and Literature survey** describes related work and our contribution.
3. Chapter 3: **Methodology and Results** explains about our method describing different approaches used in detail. It tells how we can achieve HAR.
4. Chapter 4: **Conclusion and Future** Work concludes the project's thesis along with future plans

Chapter 2

Related Work

In this domain, remarkable amount of work has been done for recognising human actions but still it is considered as a recent field of research. The huge amount of work is investigated in the literature for recognising action and activity from video sequences which basically rely on local features, space time features, global features, bag of words and bag of visual words features, keypoints descriptor based features etc. [21] (Chaaraoui et al., 2012) [22] Vishwakarma & Agrawal [23] Weinland et al., 2011 [24] Ziaeefar & Bergevin, 2015. In vision based human action recognition techniques, classification can be done at various semantic levels. [1] (Aggarwal and Ryoo, 2011) explains the classification based on structural layout of the recognition approach and categorises the recognition methodologies into two groups, namely single layer approach and hierarchical approach. These two categories are further sub divided based on the training methods and feature depiction. Classification can also be done using learning models such as support vector machine, generative models like hidden markov models etc [25] (Poppe, 2010). The type of input feature used for classification is also categorises one of the semantic levels of classification (Poppe, 2010; Weinland et al., 2010).

In HAR, a set of features are generated and are used to learn the machine which then classifies the similar actions. One can use a holistic and sparse representation of the images, then the human blob is detected and extracted from the image. This additional step is a drawback and can be removed by some pre processing. Bobick and Davis (2001) [26] use holistic representation in their motion energy images and motion history images. They extract the temporal movement and spatial location of the human in different frames and encode this information as a feature. Weinland et al. (2006) [11] further enhance the work by changing the motion energy image to a 3 dimensional motion history volume by combining different images from multiple cameras and obtaining a view independent representation of action.

Dollar, Rabaud, Cottrell, and Belongie (2005) [27] introduces Gaussian and gabor filters for representing local features as space time interest points. Subsequently various spatio temporal interest point detectors have been presented by researchers (Chakraborty, Holte, Moeslund, & Gonzàlez, 2012 [28] ; Everts, Gemert, & Gevers, 2014 [29] ; Jargalsaikhan, Little, Direkoglu,

& O'Connor, 2013 [30].These STIPs are very robust features against scale, rotation, viewpoint, illumination, background changes etc.

The human silhouette based activity recognition was presented by many researchers in which the foreground is extracted from the background to obtain the silhouette (Eweiwi, Cheema, Thurau, & Bauckhage, 2011; Gorelick, Blank, Shechtman, Irani, & Basri, 2007 [9] ; Olivieri, Conde, & Sobrino, 2012; Weinland, Boyer, & Ronfard, 2007 [31] ; Wu & Shao, 2013) [32]. The features are extracted from the silhouettes based upon the approach used. Human silhouette is also considered as fundamental entity for extracting the features. Chaaraoui and Revuelta (2014) use these silhouettes for optimised parameters by computing these parameters evolutionary. Template matching is also used by researchers for activity recognition. Weinland et al. (2007) uses the templates of region of interst and then divide the region into cells and grids so that noise effect can be reduced significantly and view oint invariance can be achieved. (Martinez-Contreras et al., 2009) uses self organising maps by clustering the motion history image templates to represent the image viewpoint.

There are many other methods which are used by researchers for activity recognition which earlier were not originally used for HAR. Optical flow, histogram of oriented gradients was originally used for tracking purposes but they work well for activity recognition also. Thurau and Hlavac (2008) use histogram of oriented gradients for representing human activity. Tran and Sorokin (2008) use shape and motion feature in combibation by using optical flow and reduce the dimension of feature vector by principal component analysis. Classification is done using nearest neibour classifier. Fathi and Mori (2008) uses optical flow method for representing spatio temporal cuboids and used generate an adaboost and use one binary classifier for learning every pair of classes to obtain multi class classification. (Weizmann from Blank et al. (2005) and KTH from Schuldt et al. (2004)) datasets are widely used for performing activity recognition.

Bag of words and bag of visual words features are used by many researchers for activity recognition and many other tasks. Wu and Shao (2013) presented a upgraded model of bag of words which is called by bag of correlated poses by using local as well as global features. They concentrate on the problem of bag of visual words which is implemented using k mean clustering and modified it for the problem of eluding geometric or structural information. It is seen that the global features are generally composed of high dimensionality and hence many methods are evolved to reduce the dimensionality of the feature. (Masoud &

Papanikolopoulos, 2003 [33] ; Olivieri et al., 2012) uses principal component analysis for reducing the dimension of feature used for activity recognition.

Local representations are used for activity recognition in recent years in which the image is represented by collection of points. These points are based on gradient information, edge information and shape based interest points. Wu et al. (2010b) and Juan and Gwun (2009) uses harris point detectors, sift and surf and susan corners for activity recognition. Laptev (2005) uses harris point detectors to represent it in 3 dimension for activity recognition. Sift points are also converted to 3 dimensional points by Kadir and Brady (2003) and Scovanner et al. (2007) [34]. (Ikizler and Duygulu, 2007)   proposed a unique approach for representing the human body as a rectangular template, then a histogram is created using circular orientations binning.

There are linear and non linear classifiers which recognise the actions into different categories. Gorelick et al. (2007) uses global features and apply non linear approach for classification by using KNN rule and Euclidean distance as a measure of classification. The nearest neighbour classifier can also be used for local features (Batra Chen, and Sukthankar (2008). The widely used method of classification is SVM which is used by many researchers for activity recognition (Schuldt, Laptev, and Caputo (2004). Laptev et al. (2007), Cao, Masoud, Boley, and Papanikolopoulos (2009), Laptev, Caputo, Schuldt, and Lindeberg (2007).

Literature Review

In this section we discuss the different approaches used in block of HAR system. Most of the action recognition approaches can be divided into various parts, mainly in Foreground detection, Feature extraction and recognition. This section describes each block independently and the various methods exploited by these blocks.

2.1 Pre-Processing

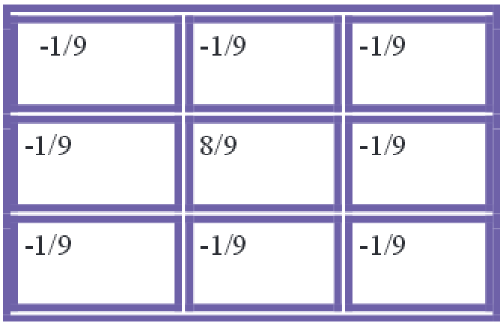
Pre-processing capacities include those operations that are ordinarily required preceding the primary information examination and extraction of data and are for the most part gathered as radiometric or geometric amendments. The pre processed images will have some noise which should be removed for the further processing of the image. Image noise is most apparent in image regions with low signal level such as shadow regions or under exposed images. There are so many types of noise like salt and pepper noise, film grains etc., All these noise are removed by using algorithms. Among the several filters, median filter is used.

2.1.1 Image Smoothing

The aim of image smoothing is to reduce the impacts of noise, spurious pixel values, missing pixel values and so forth. There are a wide range of system techniques for image smoothing. The area averaging and edge-protecting smoothing are utilized as a part of image smoothing. In neighbourhood smoothing, each point in the smoothed image, F(x,y) is obtained from the average pixel value in a neighbourhood of (x,y) in the input image. For example, if 3x3 neighbourhood around each pixel use the mask. Each pixel value is multiplied by 1/9, summed, and then the result placed in the output image. This mask is successively moved across the image until every pixel has been covered. That is, the image is convolved with this smoothing mask also known as a spatial filter or kernel. However, one usually expects the value of a pixel to be more closely related to the values of pixels close to it than to those further away.

2.1.2 Image Sharpening

The principle point in image sharpening is to highlight fine detail in the image, or to upgrade detail that has been obscured (maybe because of noise or different impacts, for example, movement). With image sharpening, the high-frequency parts are improved; this infers a spatial filter shape that has a high positive segment at the middle. A simple spatial filter that achieves image sharpening is given in Figure 2.1.



-1/9	-1/9	-1/9
-1/9	8/9	-1/9
-1/9	-1/9	-1/9

Figure 2.1: spatial filter using image sharpening

Since the sum of all the weights is zero the resulting signal will have a zero DC value i.e. the average signal value or the coefficient of the zero frequency term in the Fourier expansion. For display purposes, the value of an offset to keep the result in the 0....255 range.

2.1.3 Enhancement

The proposed system describes the information of enhancement using four types of filters such as

1. Median filter
2. Weighted Median filter
3. Adaptive filter
4. Spatial filter

For expelling high frequency parts, for example, impulsive noise, salt and pepper noise and high frequency segments. In the Enhancement stage the filters are intended to improve the presence of images, basically by sharpening Edges, corners, and line point of interest. A few of the new improvement filters incorporate noise reduction part.

Median filter: Median filtering is a nonlinear operation regularly utilized as a part of image processing to diminish "salt and pepper" noise. Median filtering is more powerful than convolution when the objective is to simultaneously diminish noise and protect edges. Median Filter can expel the noise, high frequency parts from an image without disturbing the edges and it is utilized to lessen salt and pepper noise. This strategy calculates the middle of the encompassing pixels to decide the new disparaged estimation of the pixel. A median is calculated by sorting all pixel values by their size, then selecting the middle value as the new value for the pixel.

Weighted Median filter: A weighted median filter controlled by evidence fusion is proposed for removing noise from images with contrast. It has a great potential for being used in rank order filtering and image processing. The weights of the filter are set based on intensity value of the pixels in the MRI image. Here we used four weights such as 0, 0.1, 0.2 and 0.3. If the intensity value of the pixel is 0 then consider the weight of the pixel is 0. Else if the range of pixel intensity between 1-100 then the weight is 0.1, else if the range of pixel intensity between 101-200 and the weight is 0.2, otherwise the weight of the pixel is 0.3. The above weights are multiplied with pixel intensity after that the median filter is applied for calculate weighted median filter.

Adaptive Filter: A new type of adaptive centre filter is developed for impulsive noise reduction of an image without the degradation of an original image. The image is processed using an adaptive filter. The shape of the filter basis is adapted to follow the high contrasted edges of the image. In this way the artifacts introduced by a circularly symmetric filter at the border of high contrasted areas are reduced.

Spatial Filter: Spatial filter design method is used to reduce the magnetic noise in the magnetic resonance. This filter is used to extract the external magnetic noise appearing on image and to improve the signal-to-noise ratio of the image. In spatial domain filtering, the filter is specified as 3D array. The kernel is then applied to the image via convolution or correlation using imfilter or filter2. Here, the filter for the picture elements includes a first filter that applies one filter function to the pixels in each column of the image. The partially filtered pixels are stored in matrix and then read row by row in a field interlaced order. The rows of picture elements are sent to a second filter that applies another filter function to each row. The fully filtered picture elements from the second filter are stored or converted to a matrix to display an image.

2.2 Foreground detection

2.2.1 Background Subtraction

For various computer vision applications, background subtraction (BS) is a “quick and dirty” way of localizing moving objects in a video shot by a static camera. In this perspective, motion detection is often the first step of a multi-stage computer vision system (car tracking, person recognition, wild-life monitoring, etc.). For this reason, it is usually required to be as fast and as simple as possible. Consequently, most BS methods labelm"in motion" every pixel at time t whose color is significantly different from the ones in the background. This solution has proven successful whenever the camera is rigorously static with a fixed noise-free background.

But detecting motion through background subtraction is not always as easy as it may first appear. Indeed, some videos with poor signal-to-noise ratio caused by a low quality camera, compression artifacts or a noisy environment, are likely to generate numerous false positives. False positives can also be induced by illumination changes (gradual or sudden), an animated background (waves on the water, trees shaken by the wind), or camera jitter to name a few. On the other hand, false negatives can also occur when a moving object is made of colors similar to the ones in the background (the so-called camouflage effect). With such scenarios, a simple interface difference with global threshold reveals itself as a weak solution. In order to cope with those challenges, numerous background models and distance measures bound up to different optimization schemes have been proposed in the past decade. Those methods are (at least in theory) more robust to noise and background instability than the basic background subtraction approaches. But are they really? And if they are, how much better are they? Are they suitable for real-time applications? Can they be implemented on a light weight architecture.

Background subtraction algorithm

Although different, most BS techniques share a common denominator: they make the assumption that the observed video sequence. I is made of a static background B in front of which moving objects are observed. With the assumption that every moving object is made of a color (or a color distribution) different from the one observed in B, numerous BS methods can be summarized by the following formula in equation 2.1

$$X_t(s) = \begin{Bmatrix} 1 \ if \ \ d(\, I_{s,t} \, , B_s \, > \, \tau \\ 0 \ \ otherwise \end{Bmatrix} \tag{2.1}$$

Where τ is a threshold, X_t is the motion label field at time t (also called motion mask), d is the distance between $I_{s,t}$, the color at time t and pixel s, and $\mathrm{B_s}$, the background model at pixel s.

The main difference between several BS methods is how B is modelled and which distance metric d they use. In the following subsection, various BS techniques are presented as well as their respective distance measure. Figure 2.2 shows the background subtraction process.

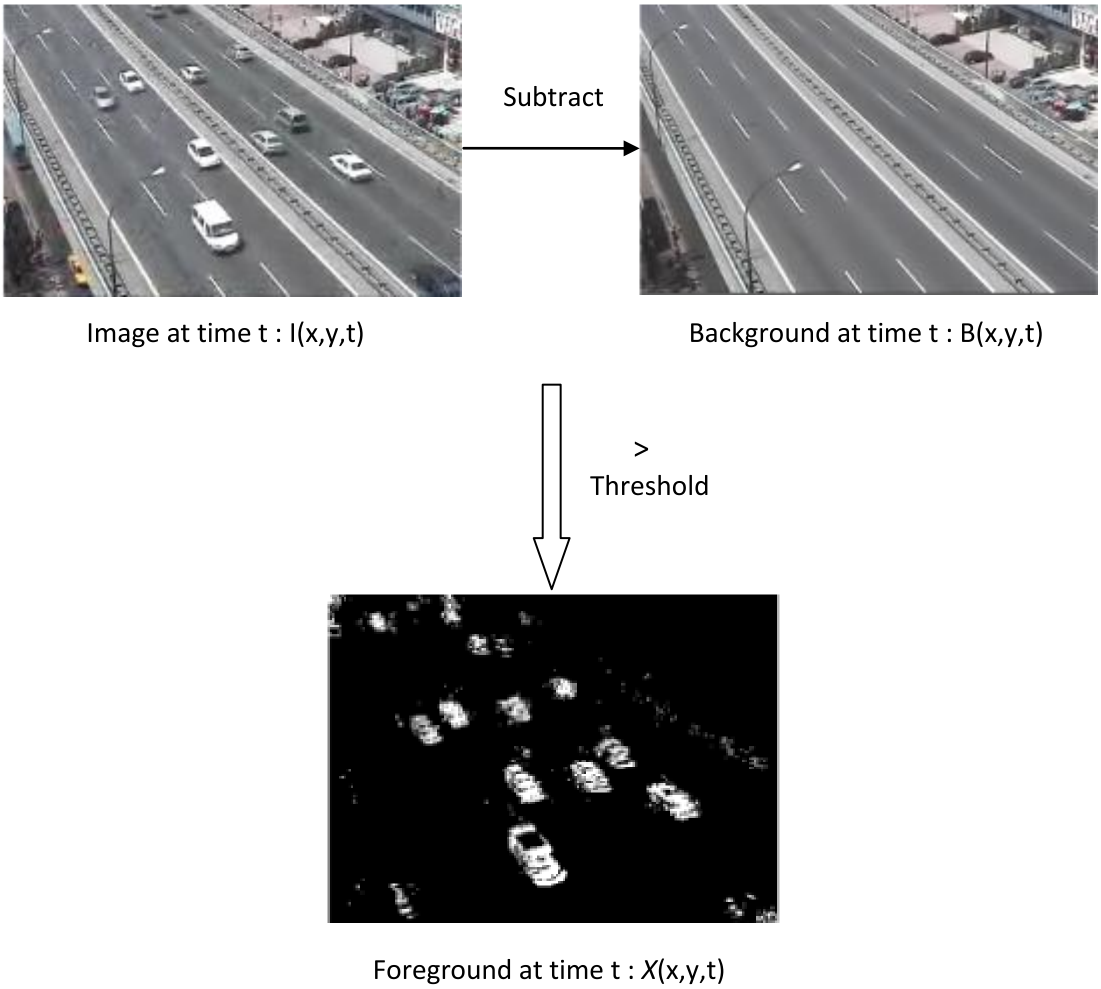

Subtract
Image at time t : I(x,y,t)
Background at time t : B(x,y,t)
>
Threshold
Foreground at time t : *X*(x,y,t)

Figure 2.2: Process showing Background subtraction

There are various ways to detect foreground such as by using Gaussian mixture model, local binary pattern, texture based segmentation etc.

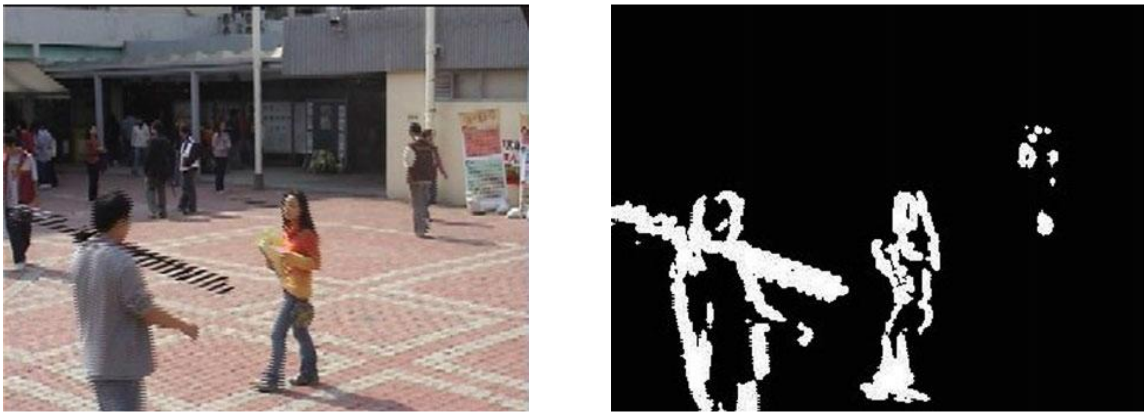
2.2.2 Temporal average filter

It is a strategy that evaluates the background model from the median of all pixels of various past images. The framework utilizes a buffer with the pixel estimations of the last frames to update the median for every image.

To show the background, the framework analyzes all images in a given time called training time. After the training period for each new frame, every pixel quantity is contrasted and compared with previously calculated input value. On the off chance that the input pixel value is inside a threshold limit, the pixel is considered to coordinate the background model and its value is incorporated into the pixbuf. Else, if the quality is outside this limit i.e. outside the threshold value, pixel is categorized to foreground area, and excluded in the buffer. This technique cannot be viewed as extremely productive on the grounds that they don't present a thorough measurable premise and requires a buffer that has a high computational expense.

2.2.3 Optical Flow

Blob detection is the primary task in video surveillance frameworks, in light of the fact that the exact procedure of segmentation and tracking depends intensely on the capacity of a framework to recognize the foreground from the background. The standard methodology is to subtract the background image, yet such a methodology comes up short when the environment becomes crowded as the background image becomes unavailable. Optical flow is one way to take care of this issue. Figures 2.3 exhibit the test results for blob discovery utilizing optical stream. In a crowded environment in which a background frame is not accessible, the customary subtraction algorithm fails to distinguish blobs, though the optical flow one performs well.



Frame for Blob Detection

Blob detection by optical flow

Figure 2.3: Foreground Detection using Optical Flow

The traditional subtraction method can process information faster than the optical flow-based approach because of its simplicity, but is dependent on the availability of a background image. This means that in crowded environments, including campuses, shopping malls, subways, and train stations, the subtraction approach will fail, whereas the optical flow-based approach will work well.

2.2.4 Texture Based Segmentation

For recognizing a human activity, foreground/background detection is a crucial task embracing various challenges including background variation, occlusion, illumination changes, noise etc. In HAR system detection of foreground produces a binary sequence of video stream data through which the object of interest can be tracked easily in subsequent frames of a video. This result in extraction of human silhouette which makes it possible to detect moving objects. Segmentation can be done using numerous methods such as background subtraction, edge based methods, shape based methods, thresholding etc. segmentation using texture properties is widely used method of differentiating uninteresting and interesting objects. This step partition the image into various sub regions allowing extracting features of the region of interest. The different texture present in a frame is useful parameter for segmentation. In HAR system dataset, we consider that the frame consists of two type of texture which makes it easier to implement. In silhouette extraction, the fundamental step is to detect the foreground by using textural property differences of human body from the whole scene. In the past, Gaussian Mixture Model (GMM) and Local Binary Pattern (LBP) based methods are widely used. A textural based segmentation technique using Gray Level co-occurrence matrix (GLCM) is proposed in. After that, various textural feature based segmentation methods have been proposed. Different techniques describing the texture properties was originally proposed in which a matrix called Gray-Level Co-occurrence Matrix is presented that describes the texture features on the basis of intensity variations in different directions.

There are different features for textures in which entropy is the most widely used parameter for describing the textural properties. Entropy measures the randomness of the gray level distribution of an image and can be expressed as equation 2.2

$$Entropy = -\sum_i \sum_j \rho(i,j)\ log\ \rho(i,j) \tag{2.2}$$

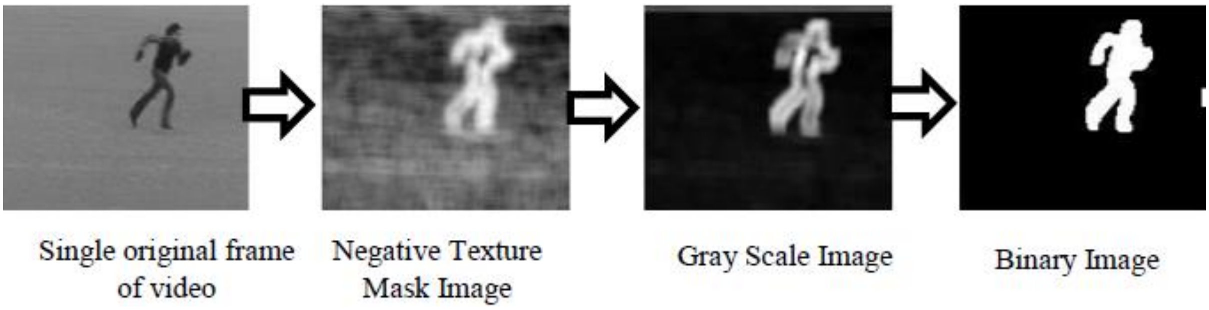


Figure 2.4: Foreground detection using texture segmentation

Where $\rho(i,j)$ represents the (i,j) th entry in normalized Gray-Level Co-occurrence Matrix. A high value of entropy means a random distribution. For representing different textures present in an image, an entropy filter is made in an image. For each pixel, a filter matrix is generated and the entropy value is determined in its 9×9 neighborhood. This filter matrix is converted to binary form by applying thresholding which gives image with white spots. Figure 2.4 describes the steps performed in segmentation of the frame to extract the silhouette of object of interest.

2.3 Feature Detectors

2.3.1 Canny Edge Detection

The need for edge recognition all in all is to essentially elicit the measure of information in an image, while protecting the basic properties to be used for further picture processing. Several algorithms exists, and this work concentrates on a specific one created by John F. Canny (JFC) in 1986. Despite the fact that it is very old, it has gotten to be one of the standard edge recognition strategies and it is still utilized as a part of research.

There are five steps involved in the canny edge detection algorithm. The steps are explained one by one with the help of examples:

Smoothing: Image blurring to remove noise.

It is unavoidable that all images taken from a camera will contain some measure of noise. To intercept that noise is mistaken for edges, noise must be diminished. Accordingly the image is first smoothed by applying a Gaussian filter. The kernel of a Gaussian filter with a standard deviation of σ= 1.4 is appeared in Equation 2.3. The impact of smoothing the test image with this filter is appeared in Figure 2.5

$$B = \frac{1}{159} \begin{matrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{matrix} \tag{2.3}$$

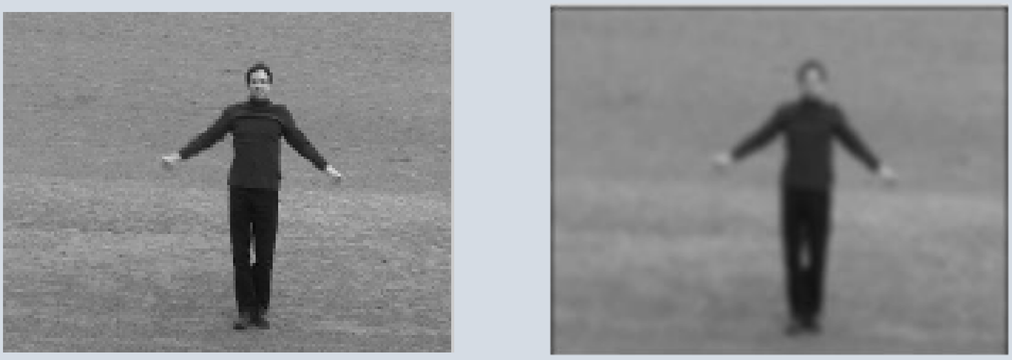


Figure 2.5: The original grayscale image is smoothed with a Gaussian filter to suppress noise

Gradient Computation: The edges should be marked where the gradients of the image has large magnitudes.

The Canny calculation essentially discovers edges where the grayscale intensity of the image changes the most. These zones are found by deciding gradients of the image. Gradients at every pixel in the smoothed image are dictated by applying what is known as the Sobel-operator. To begin with step is to approximate the gradient in the x-and y-direction individually by applying the kernels appeared in Equation 2.4

$$K_{GX} = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

$$K_{Gy} = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \tag{2.4}$$

The gradient magnitudes (otherwise called the edge strengths) can then be determined as a Euclidean separation measure by applying the law of Pythagoras as appeared in Equation. It is once in a while streamlined by applying Manhattan separation measure as appeared in Equation to lessen the computational complexity. The Euclidean separation measure has been applied to the test image. The computed edge strengths are compared to the smoothed image in Figure 2.6. The magnitude of gradients is computed using equation 2.5 and 2.6.

$$|G| = \sqrt{G_{x^2} + G_{y^2}} \tag{2.5}$$

$$|G| = |G_x| + |G_y| \tag{2.6}$$

G_x and G_y are the gradients in the x- and y-directions respectively.

It is clear from Figure 2.6, that a image of the gradient magnitudes show the edges clearly. In any case, the edges are regularly broad and thus don't show precisely where the edges are. To make it conceivable to decide this, the direction of the edges must be determined and stored as in Equation 2.7

$$\theta = arctan\frac{|G_y|}{|G_x|} \tag{2.7}$$

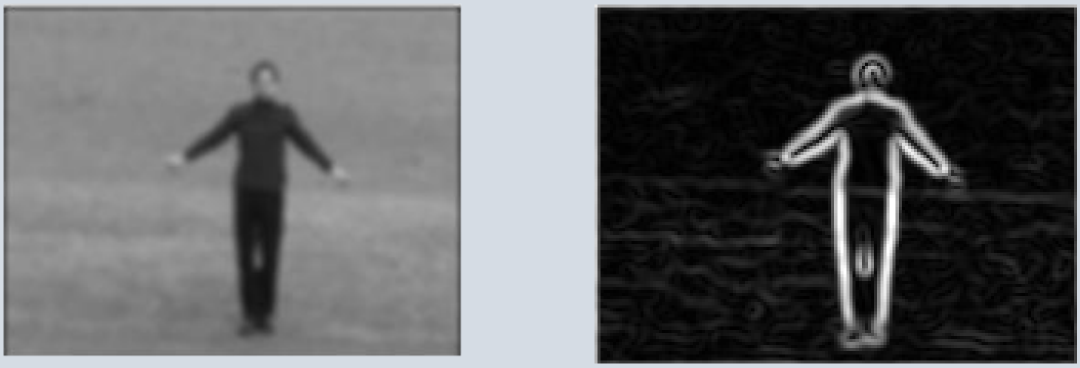


Figure 2.6: The gradient magnitudes in the smoothed image as well as their directions are determined by applying the Sobel-operator

Non-maximum suppression: Only local maxima should be marked as edges.

The motivation behind this progression is to change over the "obscured" edges in the image of the gradient magnitude to "sharp" edges. Essentially this is finished by safeguarding all neighbourhood maxima in the gradient image, and erasing everything else. The calculation is for every pixel in the gradient image:

1. Round the inclination direction θ to closest 45 degree, relating to the utilization of a 8-connected neighbourhood.
2. Compare the edge strength of the present pixel with the edge strength of the pixel in the positive and negative slope direction i.e. in the event that the slope heading is north (theta=90◦), contrast with the pixels with the north and south.
3. In the event that the edge strength of the present pixel is largest; save the value of the edge strength. If not, remove (i.e. evacuate) the quality. Figure 2.7 shows the edges after non maximum suppression.

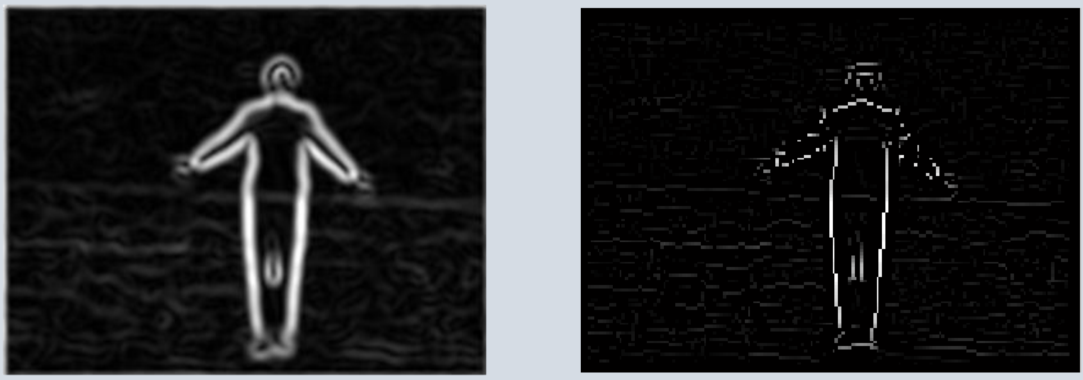


Figure 2.7: Non-maximum suppression. Edge-pixels are only preserved where the gradient has local maxima

Double thresholding: Potential edges are determined by thresholding.

The edge-pixels staying after the non-maximum suppression step are still set apart with their strength pixel-by-pixel. A significant number of these will likely be genuine edges in the image, however some may be brought on by noise or shading varieties for occurrence because of harsh surfaces. The easiest approach to observe between these eventual to utilize a limit, so that lone edges more grounded that a certain quality would be protected. The Canny edge location calculation utilizes double thresholding. Edge pixels more grounded than the high limit are set apart as solid; edge pixels weaker than the low limit are removed and edge pixels between the two edges are set apart as feeble. The impact on the test image is appeared in Figure 2.8.

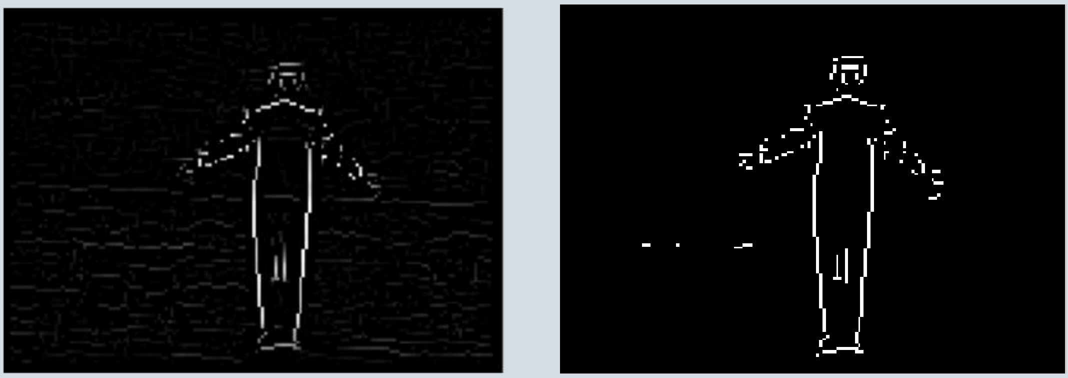


Figure 2.8: Thresholding of edges. In the second image strong edges are white, while weak edges are grey. Edges with strength below both thresholds are suppressed

Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

Solid edges are deciphered as "certain edges", and can promptly be incorporated into the last edge picture. Powerless edges are incorporated if and just on the off chance that they are associated with solid edges. The rationale is obviously that commotion and other little varieties are unlikely to bring about a solid edge (with appropriate modification of the limit levels). Subsequently solid edges will just be because of genuine edges in the first picture. The feeble edges can either be because of genuine edges or noise/shading varieties. The last sort will likely be disseminated freely of edges on the whole picture, and therefore just a little sum will be found contiguous solid edges. Frail edges because of genuine edges are a great deal more inclined to be associated straightforwardly to solid edges

2.3.2 Scale Invariant Feature Transform

This section describes in detail some of the more notable local feature detectors. A feature detector finds the points in the video where features are going to be extracted. These points are known as key points. A Key point is a point in space (x, y) that has high saliency. High saliency means that there are high amounts of changes in the neighbourhood of the point. In the spatial domain this shows as large contrast changes, yielding a Spatial Interest Point. Saliency in the temporal domain occurs when a point changes over time.

Scale Space

In the more challenging datasets, there can be large intra-class differences between videos. Videos have different camera viewpoint, scene composition, resolution, etc. This results in that the same object, e. g. a human being, in one video can have the size of hundreds of pixels, while a human in another video only takes up about 50 pixels of space. To be able to detect similar features in these videos, STIPs are detected in different temporal and spatial scales. These scales are represented as a convolution with a Gaussian blurs function, where higher values of the variance σ^2 represent larger scales. This has intuitive meaning because the more the video is blurred the more small details are lost, leaving only larger scale details behind. A video $f(x, y, t)$ is then represented by the scale-space

This stage of the filtering attempts to identify those locations and scales that is identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function. Further it has been shown under reasonable assumptions it must be based on the Gaussian function. The scale space is defined by the function in equation 2.8

$$\mathrm{L}(x, y, \sigma) = \mathrm{G}(x, y, \sigma) * \mathrm{I}(x, y) \qquad (2.8)$$

Where * is the convolution operator, $\mathrm{G}(x, y, \sigma)$ is a variable-scale Gaussian and $\mathrm{I}(x, y)$ is the input image.

Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, $\mathrm{D}(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $\mathrm{D}(x, y, \sigma)$ is then given by equation 2.9

$$\mathrm{D}(x, y, \sigma) = \mathrm{L}(x, y, \mathrm{k}\sigma) - \mathrm{L}(x, y, \sigma) \qquad (2.9)$$

To detect the local maxima and minima of D(x, y, σ) each point is compared with its 8 neighbours at the same scale, and its 9 neighbours up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

Keypoint Localisation

This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localised on an edge. This is achieved by calculating the Laplacian, value for each keypoint found in stage 1. The location of extremum, **z**, is given by equation 2.10

$$z = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} \tag{2.10}$$

If the function value at **z** is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localisation it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the key point, the key point is rejected.

Orientation Assignment

This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:

- Use the key points scale to select the Gaussian smoothed image L, from above.
- Compute:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.11}$$

- Compute:

$$\mu(x,y) = \tan^{-1}(L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)) \tag{2.12}$$

- Form an orientation histogram from gradient orientations of sample points.

- Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation
- Some points will be assigned multiple orientations.
- Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position.

Key point Descriptor

The local gradient data, used above, is also used to create key point descriptors. The gradient information is rotated to line up with the orientation of the key point and then weighted by a Gaussian with variance of 1.5 * key point scale. This data is then used to create a set of histograms over a window centred on the key point. Key point descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This result in a feature vector containing 128 elements. Figure 2.9 depicts the keypoints in hand waving activity of KTH dataset.

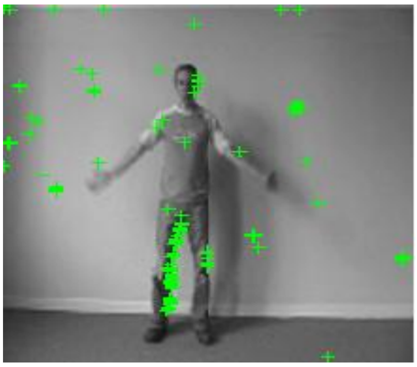


Figure 2.9: Keypoints in hand waving activity

2.3.3 Harris Corner Detector

In this section the deduction of the Harris corner identifier is displayed. The Harris corner locator is a well known interest guide identifier due toward its solid invariance to: rotation, scale, changes in illumination and image noise. The Harris corner identifier depends on the local auto-autocorrelation capacity of a signal; where the local auto-correlation functions measures the local changes of the signal with patches moved by a small amount in various directions. A discrete antecedent of the Harris detector was presented by Moravec [2]; where the discreteness alludes to the moving of the patches.

The Harris corner detector is a well-known algorithm for detection of salient regions in images. The detector operates on a windowed second moment matrix given in equation 2.13

$$\mu = g(:, \sigma_i^2\,) \;\times\; \begin{pmatrix} L_x^2 & L_xL_y \\ L_xL_y & L_y^2 \end{pmatrix} \tag{2.13}$$

Where L_ξ are derivatives of the image scale space

$$L_\xi\;(x, y, \sigma_l^2\,) = \frac{\partial L_\xi\,(x,y,\sigma_l^2\,)}{\partial \xi} \tag{2.14}$$

and σ_i is the variance used in the Gaussian kernel that serves as a window function. It is also called the integration scale. The Eigen values of μ determine if there is a corner, an edge or a "flat" area. If $\lambda 1$ and $\lambda 2$ both are large the point is a corner. If $\lambda 1 >> \lambda 2$ or $\lambda 2 >> \lambda 1$ the point is an edge, and if both Eigen values are small the area is "flat". Since the computation of Eigen values is expensive, the corners are detected as local maxima of the corner function

2.4 Feature Descriptor and Feature Representation

2.4.1 Histogram of Oriented Gradients

Histograms of Oriented Gradients [5] are a mainstream 2D descriptor initially created for person detection. The critical segments of the detector are appeared in figure 2.10. A HOG descriptor is computed using a block consisting of a lattice of cells where every cell again comprises of a grid of pixels. The quantity of pixels in a cell and number of cells in a block can be differed. The structure performing best as indicated by the original paper is 3 × 3 cells with 6 × 6 pixels.

For every cell in the square, a histogram of the inclinations in the pixels is computed. The histogram contains 9 bins with a range of either 0-180 degree or 0-360 degree , where the previous is known as unsigned and the last assigned. Each gradient votes in favour of the bin corresponding to the gradient direction, with a vote size comparing to the magnitude of gradient.

Finally, each block is concatenated into a vector v and normalized by its $L2$ norm given in equation 2.15

$$v_{norm} = \frac{v}{||v||_2^2 + \epsilon^2} \tag{2.15}$$

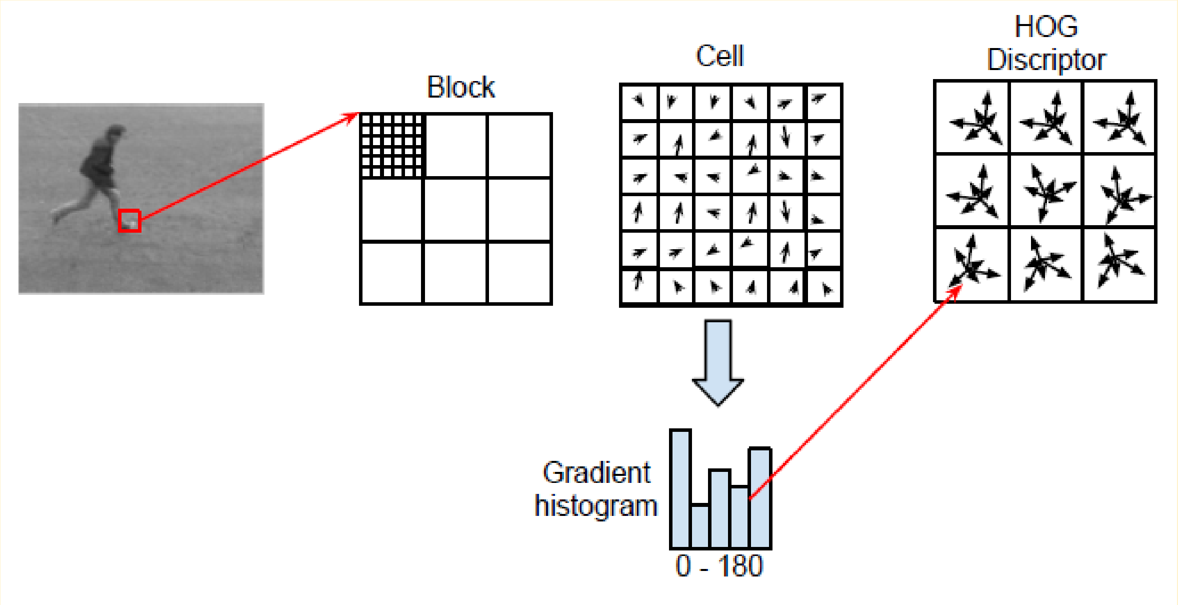


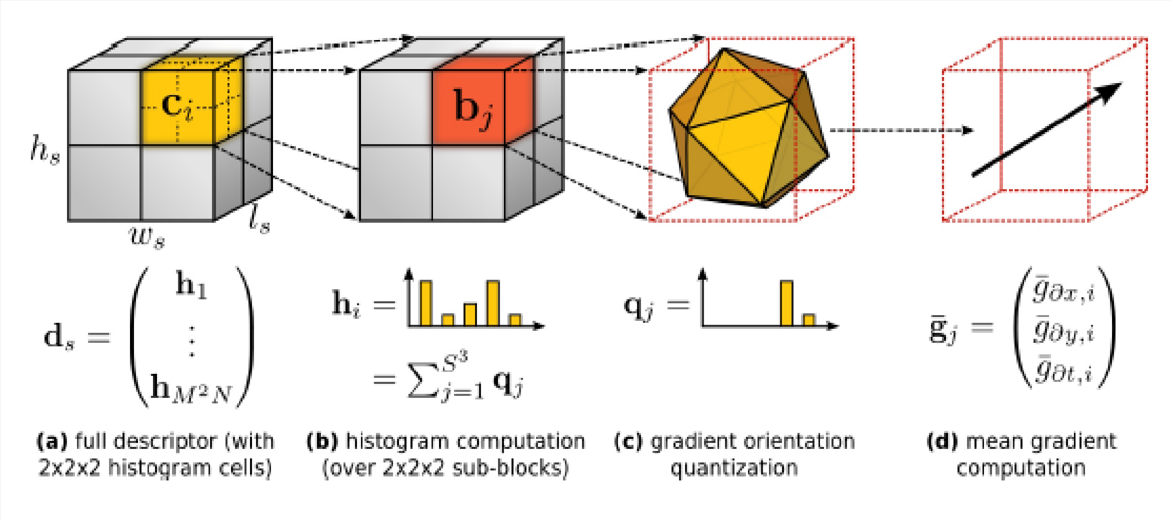
Figure 2.10: Block Diagram of HOG

Where ϵ is a small constant to prevent division by zero. The HOG descriptor is very similar to the descriptor used in SIFT [18t]. The difference is that that the SIFT descriptor is rotated according to the orientation of the interest point.

2.4.2 HOG3D

The previous methodology of utilizing HOG as a spatio-temporal feature included computing gradients in cuboids rather than in cells, however the gradient inclinations are still two-dimensional. This makes them not able to catch the temporal data in the video, which is the reason the expansion of HOF essentially enhances the outcomes.

HOG3D changes the fundamental methodology by considering the three dimensional gradient rather than the 2D angle. The gradient is figured in three measurements, and histograms are quantized into polyhedrons. This rich method for quantization is intuitive if we look at the standard method for quantizing 2D angle orientations by estimation of a circle with a polygon. Each side of the polygon corresponds to a histogram bin. By extending this to 3D the polygon turns into a polyhedron. The polyhedron utilized is an icosahedron which has 20 sides, in this manner coming about in a 20 bin histogram of 3d gradient directions. The final descriptor is acquired by concatenation of histograms into a vector, and standardization by the L2 norm. This explains in figure 2.11.


$\mathbf{c}_i$
$\mathbf{b}_j$
h_s
w_s
l_s
$\mathbf{d}_s = \begin{pmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{M^2N} \end{pmatrix}$
$\mathbf{h}_i =$
$= \sum_{j=1}^{S^3} \mathbf{q}_j$
$\mathbf{q}_j =$
$\bar{\mathbf{g}}_j = \begin{pmatrix} \bar{g}_{\partial x,i} \\ \bar{g}_{\partial y,i} \\ \bar{g}_{\partial t,i} \end{pmatrix}$

(a) full descriptor (with 2x2x2 histogram cells)

(b) histogram computation (over 2x2x2 sub-blocks)

(c) gradient orientation quantization

(d) mean gradient computation

Figure 2.11: HOG3D Block Diagram

2.4.3 Bag of Words

Bag-of-features is derived from the sack of words representation that is utilized as a part of regular natural language processing to represent a text. The thought is that the content can be represented by the events of words, dismissing the request in which they show up in the content, or to utilize the analogy in the name, put all the words in a bag and coax them out without knowing the order in which they were placed in. The after effect of the calculation is a histogram of word events, which can be more helpful for comparing texts than a direct word-by-word comparison. The bins in the histogram are called the vocabulary, and can be the complete arrangement of a wide range of words utilized as a part of the content, or as it were a subset as it may be important to filter out regular words like: the, be, to and of.

The thought is stretched out to activity recognition by Schuldt et al. [32]. Rather than a content, the histogram is presently representing a video and rather than words comprising of letters, they are currently features extracted from the videos. The vocabulary must be characterized and is frequently created by bunching a representative subset of all features in a dataset. There is no last response to the quantity of words a vocabulary ought to comprise of, yet for the most part a more complex dataset requires a bigger vocabulary. The bag-of-features histogram is processed by putting every feature into the bin of the most comparable word, frequently measured by euclidean distance. Finally the histogram is standardized with the goal that all feature vectors are comparable, regardless of the possibility that there is an alternate number of features in the recordings. Figure 2.12 shows 3 case of word histograms, where two are of the same class. The histograms are not exceptionally important to the human eye, and this absence of semantic significance is one of the drawbacks to the bag-of-features representation.

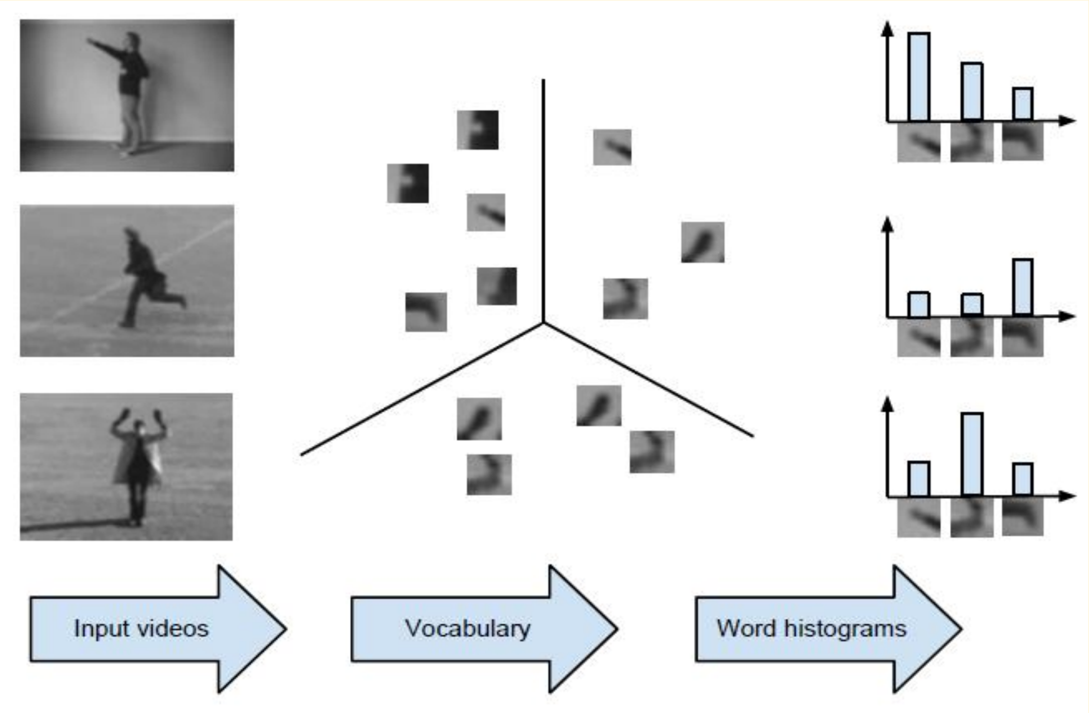


Figure 2.12: Bag of Feature representation

2.5 Supervised Learning

Supervised learning uses training data to train a classifier that encodes the differences between the classes. The training data are supervised, i. e. the labels of the samples are known, and so the classifier can learn the relationship between the features and the labels. Most of supervised classifiers require significant amounts of training data, depending on the number of classes and the complexity of the dataset.

One challenge is not to over fit the classifier to the training data, resulting in that the classifier is only able to perform well on data very similar to the training data. In the opposite case, under fitting can cause the classifier to become too general, and the accuracy will suffer.

2.5.1 Support Vector Machine

A Support Vector Machine (SVM) is a linear binary classifier that seeks to maximize the distance between the points of two classes. The solution consists of a hyper plane that separates the two classes in the best way. It is possible to extend the SVM to achieve non-linear multi-class classification.

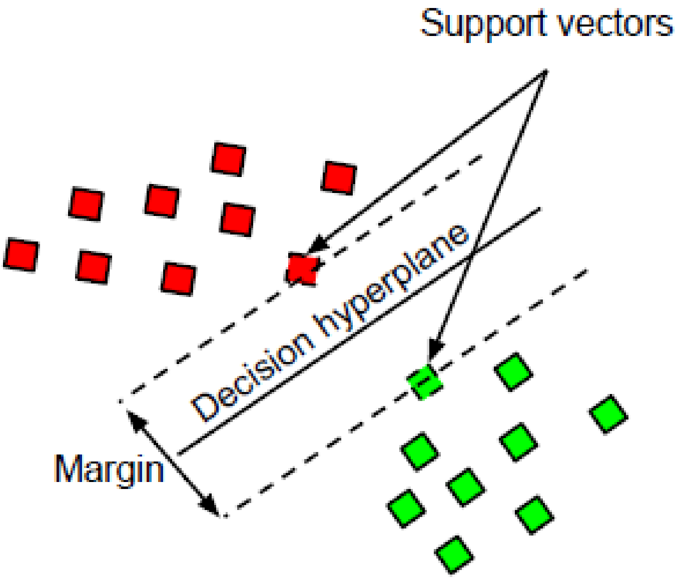


Fig 2.13: Support Vector Machine Hyperplane

Linear Separable Classes

In the simple case in figure 2.13, it is actually possible to obtain complete separation between two classes using a linear function. When the space is extended from two dimensions to n dimensions this linear function becomes a hyperplane. Many different hyperplanes could separate the points, but the desired hyperplane is the one in middle between the two point clouds, i. e. it maximizes the distance to the points. The points that influence the position of the hyperplane are the points closest to the empty space between the classes. These points are called support vectors. The distance between the hyperplane and the support vectors is the margin, so the desired hyperplane is defined as the hyperplane that maximizes the margin. We have L training points, where each input *xi* is a D-dimensional feature vector, and is one of two classes *yi* 2 {−1, +1}. The hyperplane is defined as equation 2.16

$$w \cdot x + b = 0 \qquad (2.16)$$

where *w* is the normal to the hyperplane. The objective of the SVM can then be described as finding *w* and *b* such that the two classes are separated as equation 2.17 and 2.18

$$x_i \cdot w + b \geq +1 \text{ for } y_i = +1 \qquad (2.17)$$

$$x_i \cdot w + b \geq -1 \text{ for } y_i = -1 \qquad (2.18)$$

These equations can be combined into equation 2.19

$$y_i.(x_i \cdot w + b) - 1 \geq 0 \qquad (2.19)$$

The objective is to maximize the margin, which by simple vector geometry is found to be $\frac{1}{||w||}$

This means that the optimization problem can be formulated as a minimization of $||w||$ or rather given in equation 2.20

$$\text{minimize } \frac{1}{2}\,||w||^{\,2}$$

$$\text{subject to } y_i.(x_i \cdot w + b) - 1 \geq 0 \qquad (2.20)$$

as this makes it possible to perform quadratic programming optimization later on. By introducing Lagrange multipliers α, the dual problem can eventually be reached as in equation 2.21

$$\text{maximize } \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\,\alpha^T H \alpha$$

$$\text{subject to } \alpha_i \geq 0 \qquad (2.21)$$

This problem is solved using a quadratic programming solver, yielding a solution for α and thereby *w* and *b*. Points can now be classified using the decision function in equation 2.22

$$Y' = sgn\,(w \cdot x' + b) \qquad (2.22)$$

Non Separable Points

In the formulation used above, it is assumed that the two classes are completely separable by a hyperplane, but this is often not the case. The points are often entangled in a way that makes it impossible to separate them, but it is still desired to have a classifier that makes as few mistakes as possible.

This is achieved by introducing a positive slack variableξ_i, $i = 1 . . .L$. This slack allows the points to be located on the "wrong" side of the hyperplane as seen in the modified expressions as equation 2.23

$$
\begin{aligned}
&x_i \cdot w+b \geq +1 - \xi_i \qquad \text{for } y_i = +1 \\
&xi \cdot w+b \geq\ -1 + \xi_i \qquad \text{for } yi = -1 \\
&\xi_i \geq 0
\end{aligned}
\tag{2.23}
$$

The minimization problem is reformulated as equation 2.24

$$
\text{Minimize } \frac{1}{2}\ ||w||^{\,2} + \mathrm{C}\sum_{\mathrm{i}=1}^{\mathrm{L}} \xi_i
$$

$$
\text{subject to } \ y_i.(x_i \cdot w + b) - 1 + \xi_i \geq 0 \tag{2.24}
$$

Where the parameter C controls how much the slack variables are punished.

Extending to non-linear

In the examples above, a linear decision function was considered, but in many cases, the data are non-linear. A way to classify the non-linear data is to map the data onto another space where the data are separable. In the example shown in figure 2.14a, a linear classifier would not yield a good result, but by mapping the data to polar coordinates as shown in 2.14b, the data can be completely separated by a linear classifier.

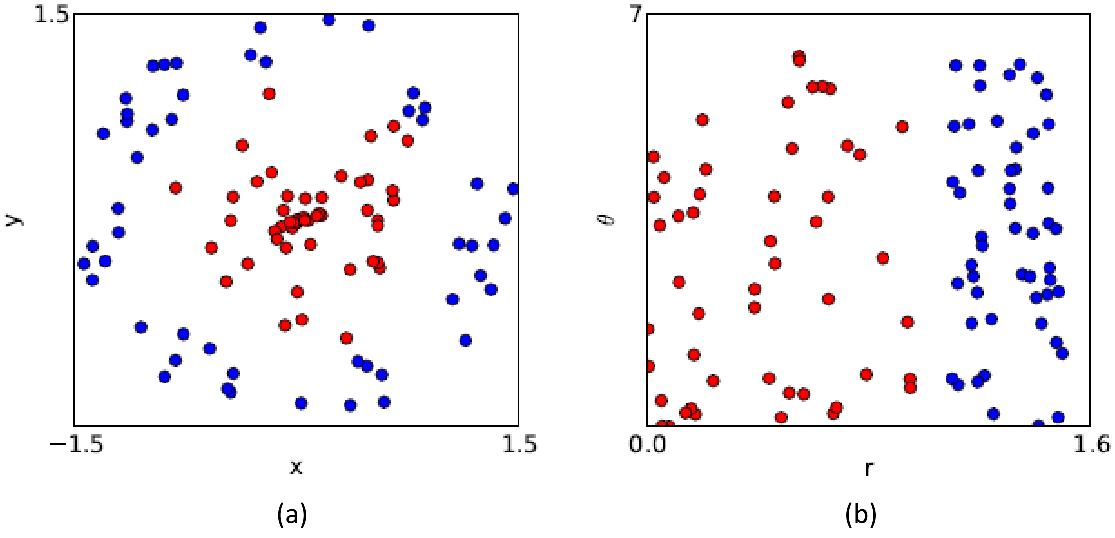


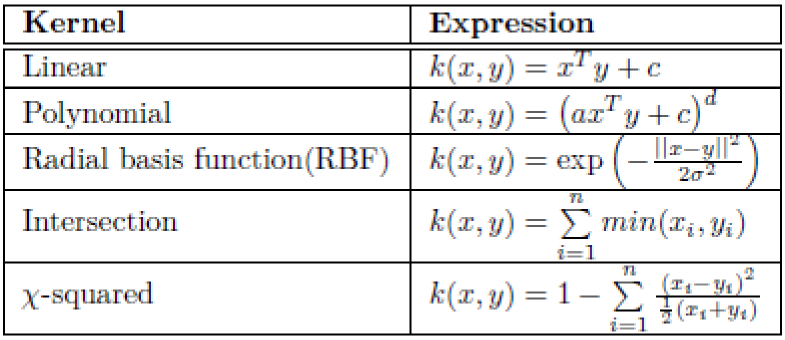
Fig 2.14: (a) Data points shown in cartesian coordinates,

And

(b) Data points shown in polar coordinates

Table 2.1 shows some popular choices of SVM kernels. The intersection and χ squared kernels are especially useful when it comes to bag-of-features classification because they provide a good way to capture the difference between histograms which is the common way to represent a video in bag-of-features.

Table 2.1: Examples of kernel functions



Kernel	Expression
Linear	$k(x,y) = x^T y + c$
Polynomial	$k(x,y) = \left(ax^T y + c\right)^d$
Radial basis function(RBF)	$k(x,y) = \exp\left(-\frac{\|\|x-y\|\|^2}{2\sigma^2}\right)$
Intersection	$k(x,y) = \sum_{i=1}^{n} min(x_i, y_i)$
χ-squared	$k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i-y_i)^2}{\frac{1}{2}(x_i+y_i)}$

Multi-class approach

The above-mentioned SVM classifiers have all considered the binary case of labelling a video as either one class or the other, but this can of course be extended to the general case of n classes. Two common approaches are: one-vs-all and one-vs-one.

In one-vs-all, each classifier is trained using training data from one class, vs. the training data from all other classes resulting in n classifiers for n classes. In classification, a point is assigned to class that has the highest output of the decision function. The classifiers have to be trained in way that results in the same range of their output to make the decision functions comparable.

In one-vs-one a classifier is constructed for each combination of two classes which results in $n(n - 1)/2$ classifiers. In the classification step, each point casts a vote for one of the classes in each of the classifiers and in the end picks the class having the highest number of votes. In the event of a tie, the class being assigned is usually arbitrary.

2.5.2 Hidden Markov Model

The Hidden Markov Model(HMM) is a powerful statistical tool for modelling generative sequences that can be characterised by an underlying process generating an observable sequence. HMMs have found application in many areas interested in signal processing, and in particular speech processing, but have also been applied with success to low level NLP tasks such as part-of-speech tagging, phrase chunking, and extracting target information from documents. Andrei Markov gave his name to the mathematical theory of Markov processes in the early twentieth century[3], but it was Baum and his colleagues that developed the theory of HMMs in the 1960s[2].

A hidden Markov model (HMM) is a triple (π,A,B).

$\Pi = (\pi)$ The vector of initial state probabilities;

$A = (a_{ij})$ the state transition matrix;

$B = (b_{ij})$ the confusion matrix;

$P_r \left(x_{i_t} \,\middle|\, x_{j_{t-1}} \right)$

$P_r \left(y_i \mid x_j \right)$

Each probability in the state transition matrix and in the confusion matrix is time independent that is, the matrices do not change in time as the system evolves. In practice, this is one of the most unrealistic assumptions of Markov models about real processes.

We want to find the probability of an observed sequence given an HMM - that is, the parameters (π,A,B) are known. Consider the weather example; we have a HMM describing the weather and its relation to the state of the seaweed, and we also have a sequence of seaweed observations. Suppose the observations for 3 consecutive days are (dry, damp, soggy) - on each of these days, the weather may have been sunny, cloudy or rainy. We can picture the observations and the possible hidden states as a trellis in figure 2.15

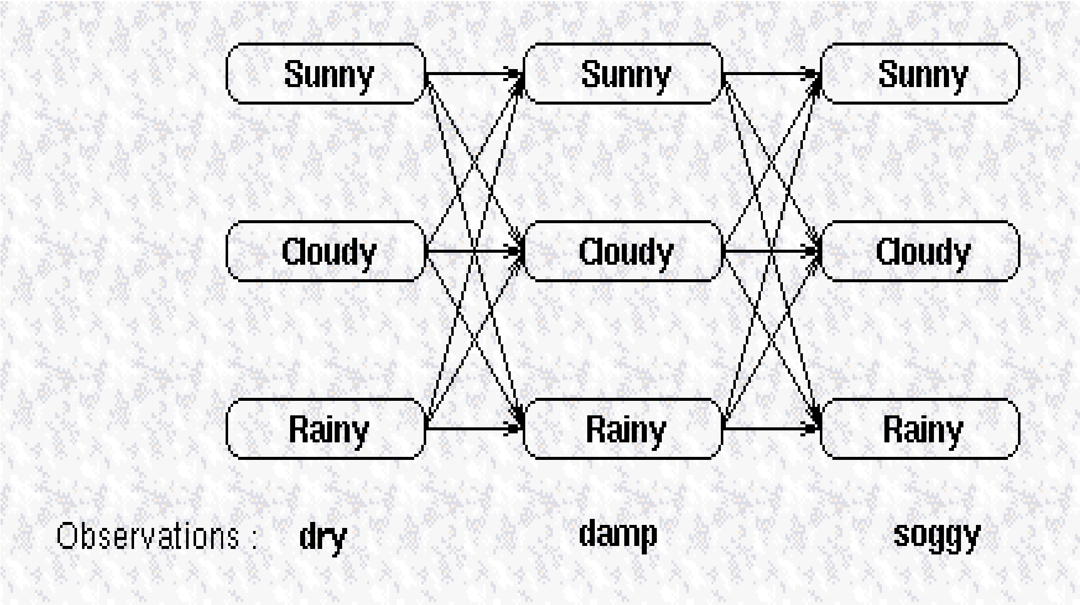


Figure 2.15: Trellis Diagram of HMM

When implementing a HMM, floating-point underflow is a significant problem. It is apparent that when applying the Viterbi or forward algorithms to long sequences the extremely small probability values that would result could underflow on most machines. We solve this problem differently for each algorithm:

Viterbi underflow

As the Viterbi algorithms only multiplies probabilities, a simple solution to underflow is to log all the probability values and then add values instead of multiply. In fact if all the values in the model matrices (π,A,B) are stored logged, then at runtime only addition operations are needed.

Forward algorithm underflow

The forward algorithm sums probability values, so it is not a viable solution to log the values in order to avoid underflow. The most common solution to this problem is to use scaling coefficients that keep the probability values in the dynamic range of the machine, and that are dependent only on t. The coefficient c_t is defined as in equation 2.23

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_i(i)} \qquad (2.23)$$

And thus the new scaled value for α becomes:

$$\alpha\hat{}_t = c_t \times \alpha_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_i(i)} \qquad (2.24)$$

A similar coefficient can be computed for $\beta\hat{}_t(i)$.

Chapter 3

Methodology

In this work, we have proposed a new hybrid method for feature calculation for recognition of human activity on standard datasets used for HAR. The novelty of this work is in the use of two features for classification namely, spatial distribution gradients and spatio temporal keypoints. Then these features are used to train SVM and HMM for accurate recognition. Then the accuracy of the two classifiers based on the hybrid features are calculated and compared.

The philanthropy given by this work for efficient representation of human actions is in feature extraction. Firstly, a entropy based texture segmentation is used for foreground detection in which human activity silhouettes are extracted from the video sequences. The various silhouettes from the activity are used to get the average energy image (AEI) features for each activity. The average energy image is a unique feature for an activity in such a way that the actions performed by the same person i.e. intra-class variations are reduced and the actions performed by different persons are maximized, such that more discriminative information can be extracted for classification.

The pyramid histogram of oriented gradients (PHOG) descriptor is applied on average energy images for computation of feature vector. The PHOG is computed for three levels and instead of going to the fourth level, the previously computed three level parameters are concatenated for reducing the dimension of feature vector. This feature vector is called as spatial distribution gradients. SDG have a disadvantage of finding features in single orientation. To overcome this problem, spatio temporal keypoints are applied to the different frames of the videos for effective representation of human activity. They extract information by changing scale and orientation. This hybrid feature is exploited for each human activity category of dataset, on which support vector machine is applied for classifying the activity performed in a video. The same feature vectors are used to train hidden markov model and the accuracy of HMM is compared with that of SVM.

The block diagram showing the flow of the above work is given in Figure 3.1. The block diagram depicts the various entities in which each entity represents a process through which activity recognition is performed.

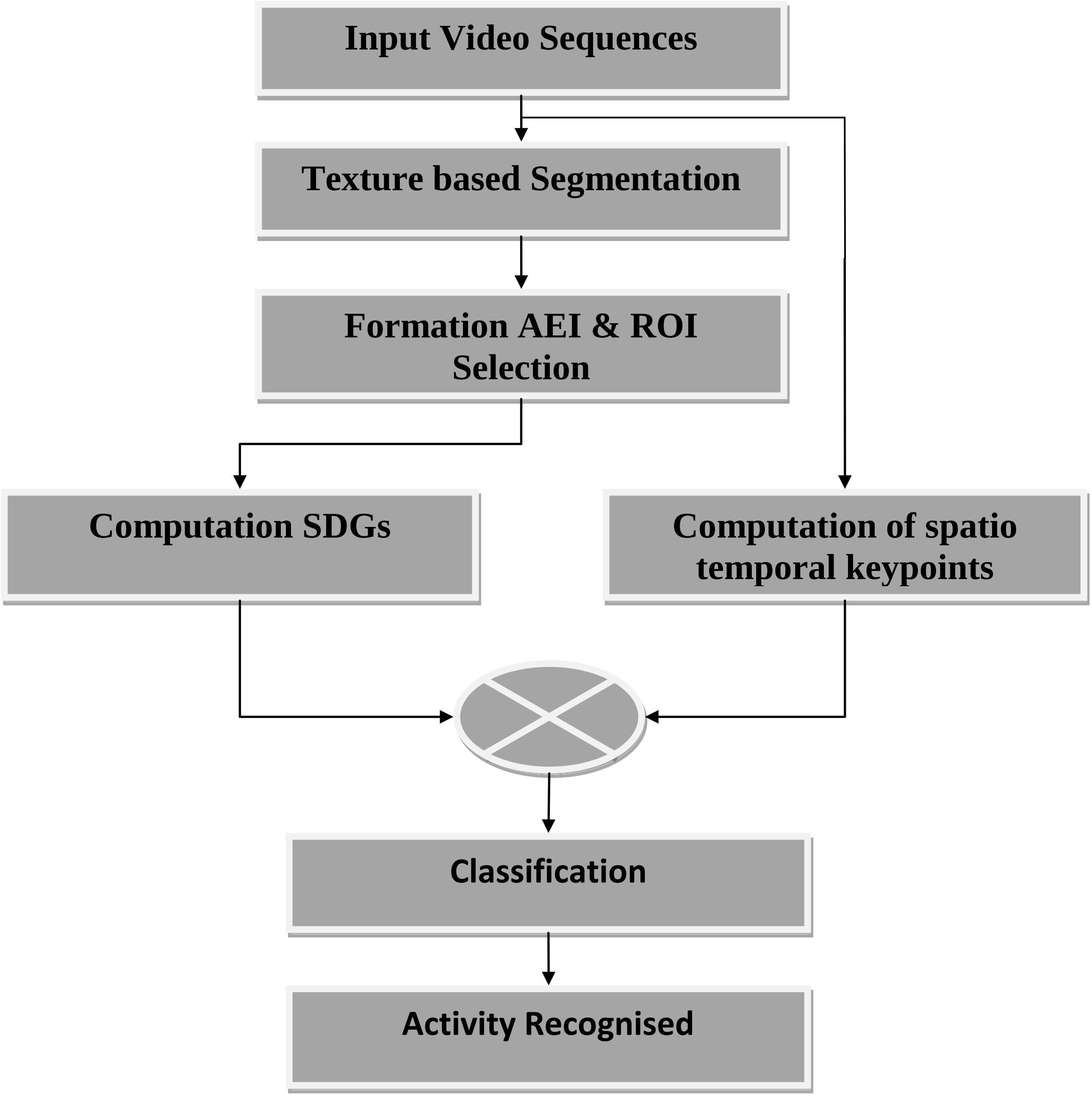

Input Video Sequences
Texture based Segmentation
Formation AEI & ROI Selection
Computation SDGs
Computation of spatio temporal keypoints
Classification
Activity Recognised

Figure.3.1. Flow diagram of proposed framework

3.1 Input Video Sequences

This thesis mainly focuses on vision based activity recognition in which video cameras are the primary source through which data is acquired. In our work, we work upon four datasets, Weizmann, KTH, ballet and IXMAS dataset. These datasets consists of daily life activities like jumping, running, walking, boxing, bending, waving etc. These are used as standard datasets for human activity recognition so that the accuracy of our work can be compared with other state of the art methods. These datasets are processed for each and every class. The videos contained in both the datasets are used for both learning and validation purposes.

KTH dataset

Schuldt et al. introduced the KTH dataset in 2004 and has various challenges in comparison to the Weizmann dataset [8]. This dataset comprises of six activities shown in figure 3.2, namely; ‘‘Hand-Clapping,’’ ‘‘Hand-Waving,’’ ‘‘Jogging,’’ ‘‘Jumping,’’ ‘‘Running,’’ and ‘‘Walking’’. There are 100 videos in each activity in different conditions. These sequences are recorded with a static camera in uniform background at a frame rate of 25 frames per second and having a spatial resolution of 160×120. There is a significant movement in camera while recording due to which segmentation is a challenging task. Texture based segmentation is used in this context.

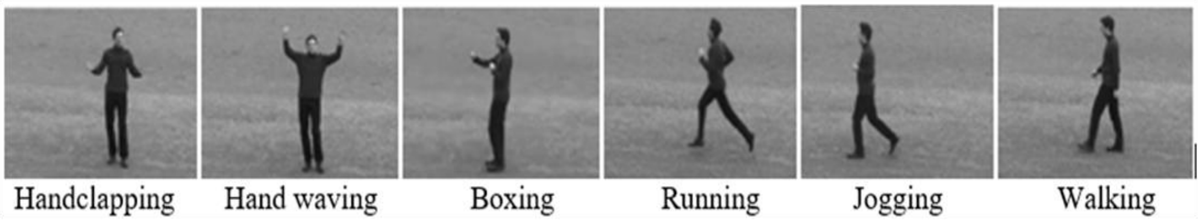


Figure 3.2: KTH Dataset

Weizmann dataset

Gorelick et al. introduced this dataset in 2007 [9]. This dataset comprises of 10 activities, namely run, walk, jump, gallop-sideways, jack, pjump, wave1, wave2, skip, bend. Each video sequence has a frame rate of 25 frames per second at a spatial resolution of 144×180. There are a total of 9 videos in each activity performed by 9 different actors with a total of 90 video sequences. These sequences are recorded with a static camera. This dataset is less challenging as compared to KTH dataset. Figure 3.3 depicts the various action of the dataset.

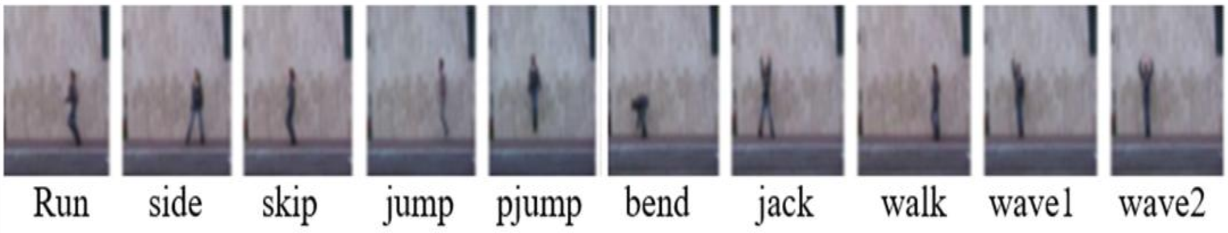


Figure 3.3: Weizmann dataset sample frames.

Ballet Dataset

The Ballet Movement activity information set (Fathi, 2008) is one of the complex human activity information sets [10]. This information set comprise of eight Ballet Movements performed by three on-screen characters and these movements are named as Hopping , Jump, Left-to-Right Hand Opening, Leg Swinging, Right-to-Left Hand Opening, Standing with Hand Opening, Stand Still and Turning. The information set is exceptionally challenging because of the extensive measure of intra-class dissimilarities in terms of spatial and temporal scale, speed, and attire. Figure 3.4 shows the eight actions of ballet movements.

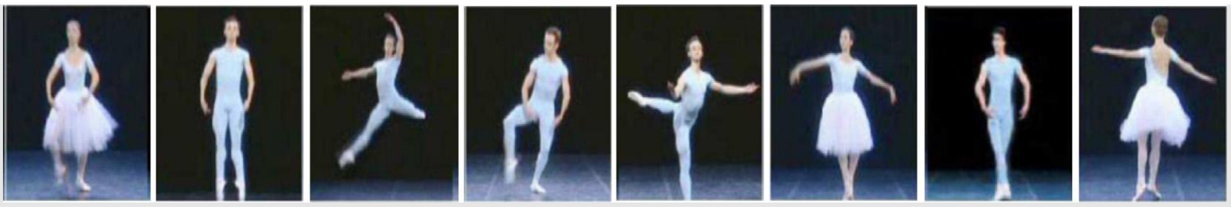


Figure 3.4: Ballet dataset actions (left to right) : Hopping, jumping, left to right hand opening, leg swinging, right to left hand opening, standing hand opening, standing still, tuning right.

IXMAS Dataset

With the reason for developing the experimentation of our technique to a more troublesome dataset with more camera perspectives, we have picked the IXMAS dataset which is main stream among human activity recognition techniques that are particularly intended for multiview recognition. The INRIA Xmas Motion Acquisition Sequences (IXMAS) dataset (Weinland et al., 2006) [11] incorporates multi-view information furthermore, is particularly gone for perspective invariance testing. It gives 390×291 px resolution images from five diverse points including four sides and one top-view camera. An arrangement of 12 on-screen characters have been recorded performing 14 distinctive activities (check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, throw over head and throw from bottom up) 3 times each, subsequent in a dataset with more than

2000 arrangements. This benchmark exhibits an expanded trouble since subjects were asked to openly pick their position and introduction. Figure 3.5 shows some of the actions at different camera positions.

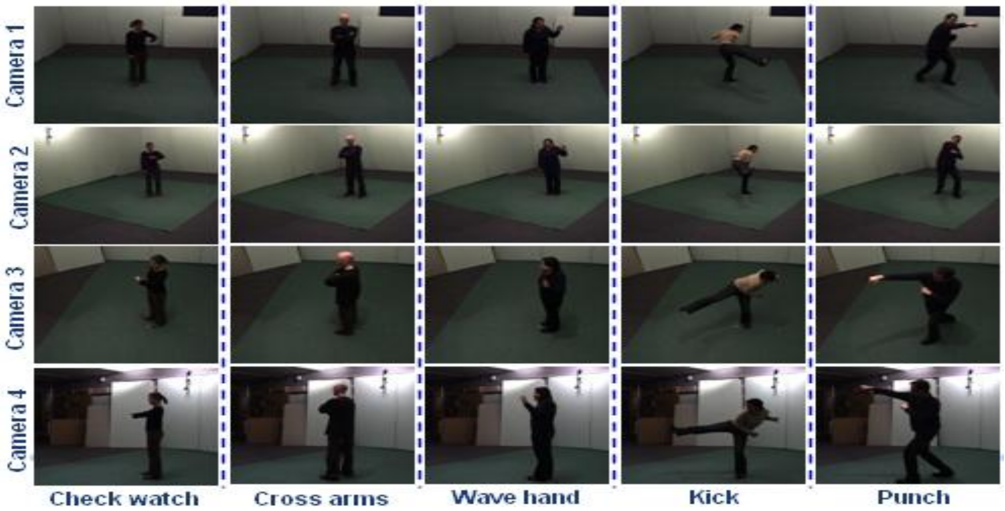


Figure 3.5: IXMAS action sequences at different camera views.

3.2 Silhouette Extraction

For recognizing a human activity, foreground/background detection is a crucial task embracing various challenges including background variation, occlusion, illumination changes, noise etc [5]. In HAR system detection of foreground produces a binary sequence of video stream data through which the object of interest can be tracked easily in subsequent frames of a video. This result in extraction of human silhouette which makes it possible to detect moving objects. Segmentation can be done using numerous methods such as background subtraction, edge based methods, shape based methods, thresholding etc. segmentation using texture properties is widely used method of differentiating uninteresting and interesting objects. This step partition the image into various sub regions allowing extracting features of the region of interest. The different texture present in a frame is useful parameter for segmentation. In HAR system dataset, we consider that the frame consists of two type of texture which makes it easier to implement. In silhouette extraction, the fundamental step is to detect the foreground by using textural property differences of human body from the whole scene. In the past, Gaussian Mixture Model (GMM) and Local Binary Pattern (LBP) based methods are widely used. A textural based segmentation technique using Gray Level co-occurrence matrix (GLCM) is proposed in [35]. After that, various textural feature based segmentation methods have been proposed [36]. Different techniques describing the texture properties was originally proposed in [37] in which a matrix called Gray-Level Co-occurrence Matrix is presented that describes the texture features on the basis of intensity variations in different directions.

There are different features for textures in which entropy is the most widely used parameter for describing the textural properties. Entropy measures the randomness of the gray level distribution of an image and can be expressed as in equation 3.1

$$Entropy = -\sum_i \sum_j \rho(i,j)\, log\, \rho(i,j) \qquad (3.1)$$

Where $\rho(i,j)$ represents the (i,j) th entry in normalized Gray-Level Co-occurrence Matrix. A high value of entropy means a random distribution. For representing different textures present in an image, an entropy filter is made in an image. For each pixel, a filter matrix is generated ant the entropy value is determined in its 9×9 neighborhood. This filter matrix is converted to binary form by applying thresholding which gives image with white spots. Figure 3.6 describes the steps performed in segmentation of the frame to extract the silhouette of object of interest.

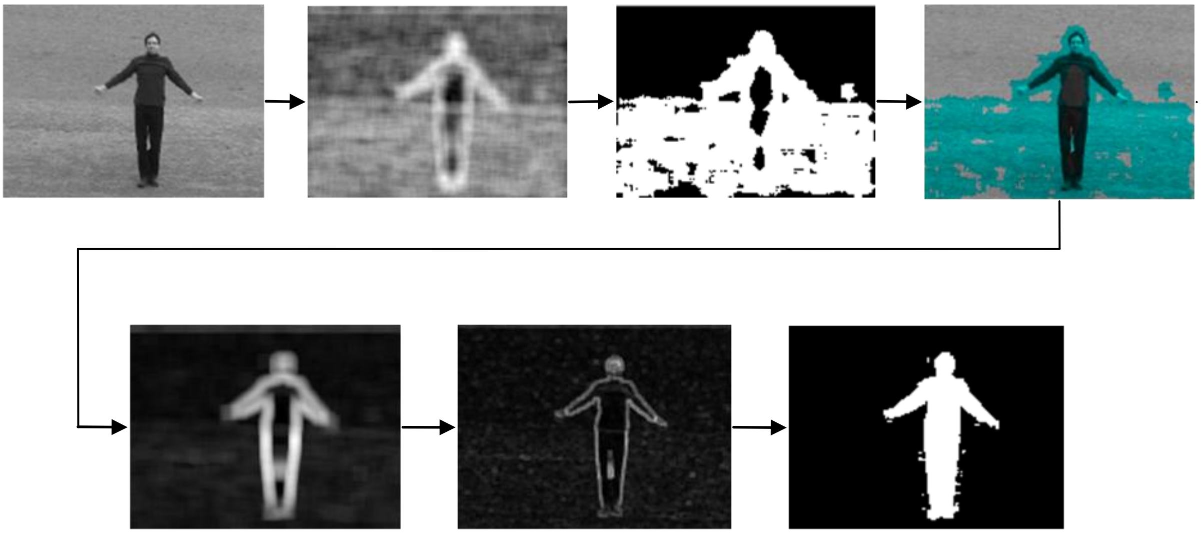


Figure 3.6: The workflow of silhouette extraction: (a) Original frame of video, (b) Texture Mask, (c), (d) Images selected by rough mask, (e) Silhouette extracted Image from original frame, (f) image showing boundary between two textures (g) Final segmentation result

The steps performed in segmenting the image by using the textural properties of the image are explained as:

Step 1: Read the video. In that video, read each and every frame. For each frame perform texture segmentation.

Step 2: Create texture mask of the frame using entropy filter. The function entropyfilt returns an array where each output pixel contains the entropy value of the 9-by-9 neighbourhood around the corresponding pixel in the input image. Entropy is a statistical measure of randomness.

Step 3: Create rough mask using thresholding. A threshold is selected such that it is roughly the intensity value of pixels along the boundary between the textures.

Step 4: Silhouettes are extracted from the image after creating the rough mask for another texture.

Step 5: A boundary is created between the two textures to ensure that the texture segmentation is done appropriately.

Step 6: Display segmentation results. The final results showing the segmented images using texture based segmentation is displayed.

3.3 Average Energy Image (AEI) feature computation

There are various methods which are used for representing human activities. Some of them are feature descriptors, bag of features representations, local features, feature detectors etc [38]. In this paper, average energy image features are exploited. The human body silhouettes are extracted for each video sequence of various activities. Let S = {S1, . . .Si, . . .Sn} be a set of human silhouettes from a video clip, where Si is the ith binary human silhouette and n represents number of frames.

Having obtained silhouette image set, each video sequence is represented by an average energy image (AEI) as a feature vector which is a gait energy image extension. The average energy image is evaluated by equation 3.2

$$AEI(p,q) = \frac{1}{n}\sum_{i=1}^{n} S\,i(p,q) \qquad (3.2)$$

where p and q are coordinates of S_i.

The average energy images are cropped to determine the region of interest (ROI) to reduce the dimension of the image. The ROIs are resized to 64×38 for better representing the average energy images. Average energy features are exploited such that it minimizes the intra-class variations. Figure.3. depicts the ROIs and AEIs of various video clips of an activity.

Consider Figure 3.7(c), the gray values are due to averaging the different silhouettes to get one image. The gray values of the pixels in the hand swing region show the quantity of motion of limbs occurring at those points in an activity. Meanwhile the white pixels in the torso, head and legs regions indicate the overall built of the person and average pose during the action performance. Thus, AEI captures both structural and motion characteristics of an action.

Following are the advantages of AEI representation of actions:-

- AEI represents major shapes of silhouettes and their changes over the activity cycle.
- AEI represents human motion sequence in a single image while preserving the average temporal information.
- Since the AEI is obtained by an averaging operation, it reduces the noise effects of background subtracted noisy silhouettes.
- With the AEI based representation there is no need for time alignment of activity frames as with the approaches based on matching features from key poses of an activity.
- AEI is a compact 2D representation of average 3D spatio-temporal information.
- AEIs save storage space.

ROI selection:

In figure 3.7, the average energy image computation is depicted. 3.7(a) represents the action clips in a hand waving activity of KTH dataset. In this figure, some of the clips are depicted for representation purpose. In actual many images are used for calculation of aei image. In 3.7(b), the corresponding segmented silhouettes are represented for each clip. This is done using texture segmentation. These silhouettes are summed and then averaged to get the aei image which is shown in 3.7(c). In 3.7(d), the region of interest is extracted so that the image size can be reduced and the size of feature vector can also be reduced.

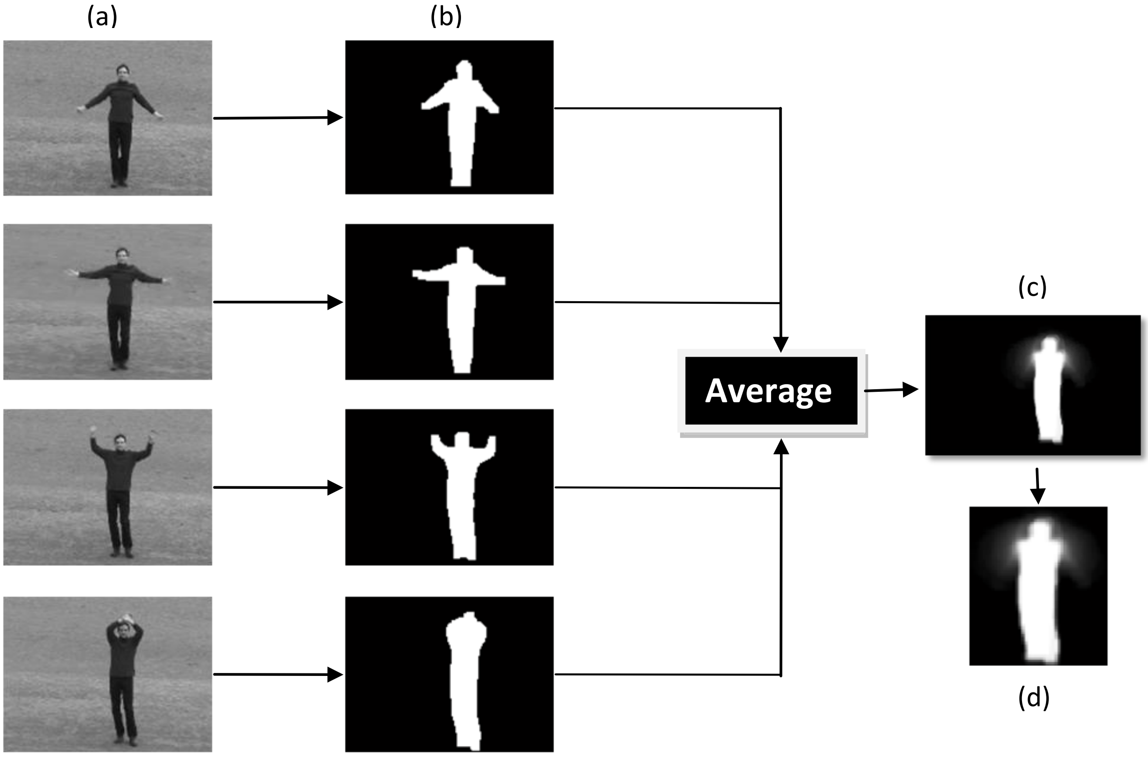


Figure 3.7: Flow Diagram depicting AEI feature computation: (a) represents the action clips performed in a hand waving activity. (b) Shows the corresponding segmented images of each clip. (c) Average energy image. (d) Region of interest (ROI).

The average energy images are cropped to determine the region of interest (ROI) to reduce the dimension of the image. The ROIs are resized to 64×38 for better representing the average energy images. Average energy features are exploited such that it minimizes the intra-class variations.

In figure 3.8 the pixel intensity values of the AEI image for hand waving activity is depicted. The AEI image is cropped to get the region of interest. The ROI image is then reshaped to convert it from two dimensional to 1 dimensional vector. This vector is then plotted to form the curve shown above. In this curve, the left and right portion indicates the gray values where amount of motion of hands is occurring during the hand waving activity. During the activity, the main body of the actor performing the hand waving action is still, i.e. only hand movement is present. Hence the middle portion of the curve represents the pixels values at those points which are not in motion. The still body points have pixel intensity of value 1, whereas the hands have different pixels values ranging from greater than to 0 to less than 1.

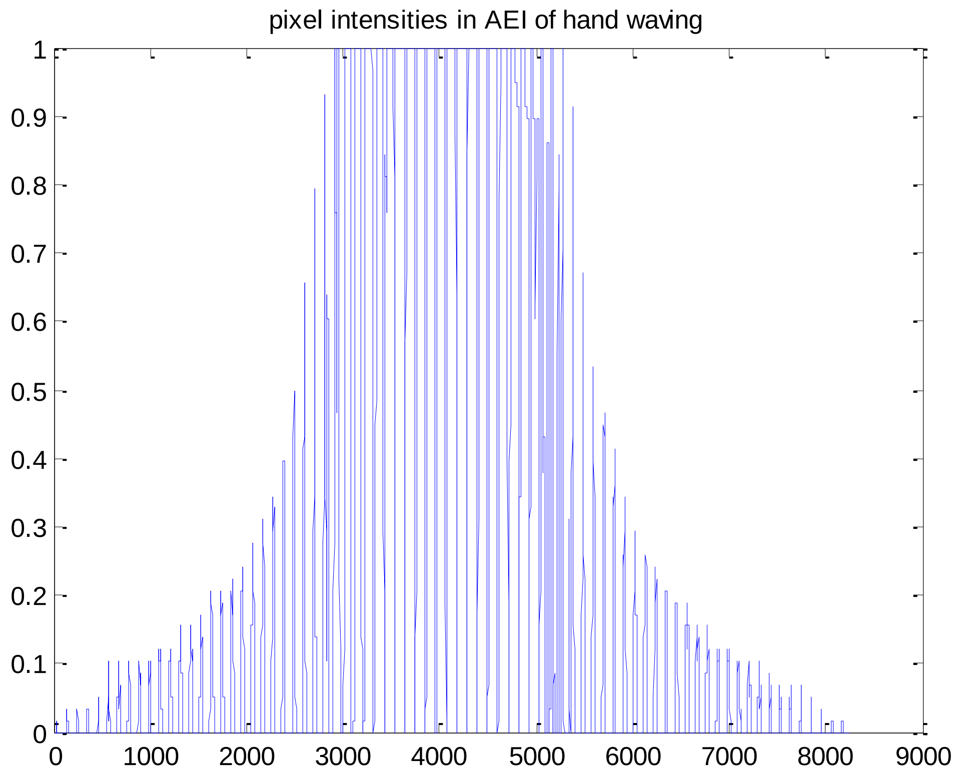
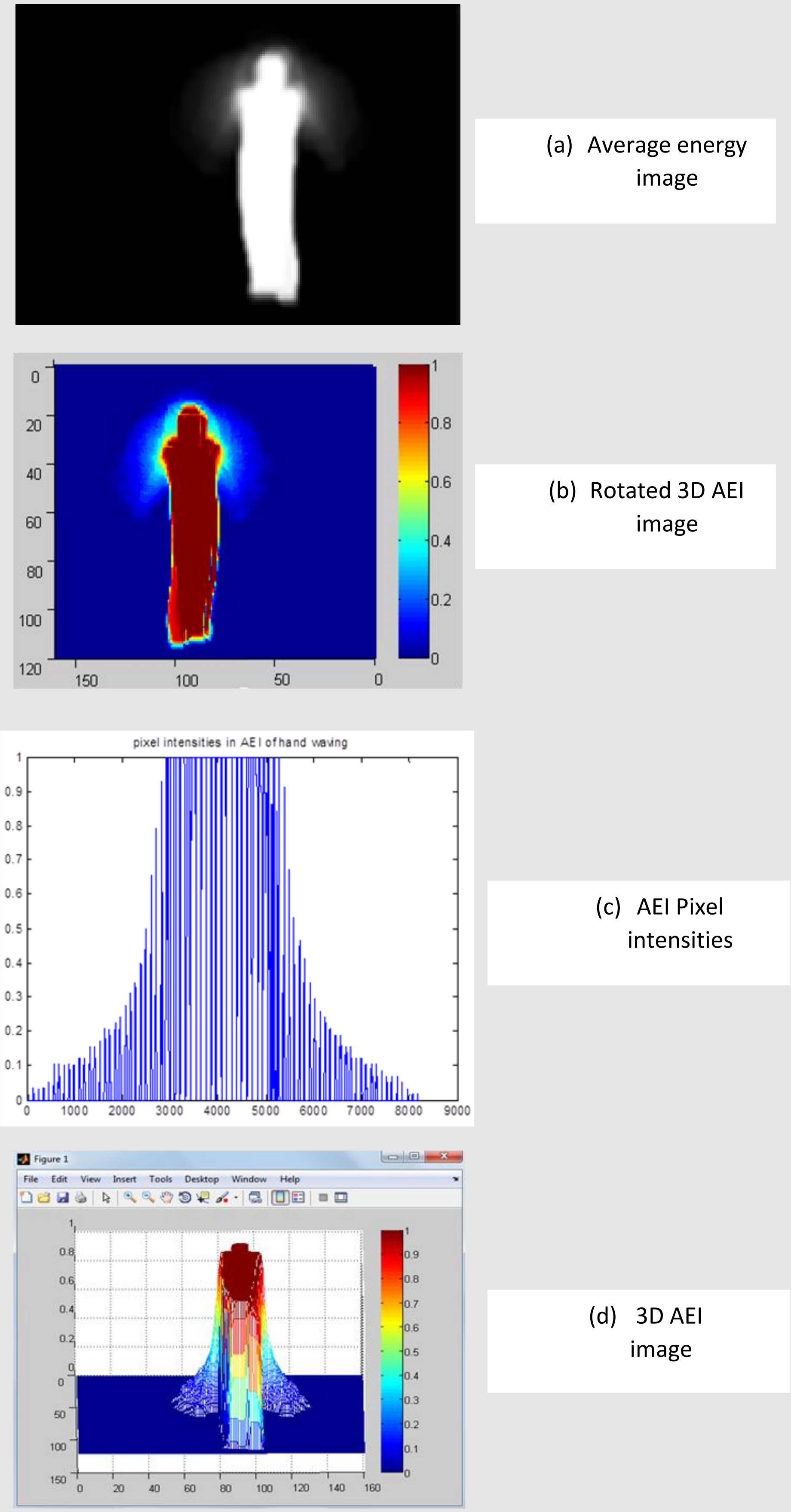


Fig 3.8: Pixel intensity values of AEI image for hand waving activity

For HAR system in which the datasets having different action classes, for each action class and for each video, one AEI image is formed. Due to this, the data representation is reduced as a video containing large number of frames is characterized by only one AEI image. Hence data reduction is possible by using AEI images. Figure 3.9 depicts the 3D and colored version of AEI images.



(a) Average energy image

(b) Rotated 3D AEI image

(c) AEI Pixel intensities

(d) 3D AEI image

Figure 3.9: AEI Image Representation

AEI Images of IXMAS Dataset

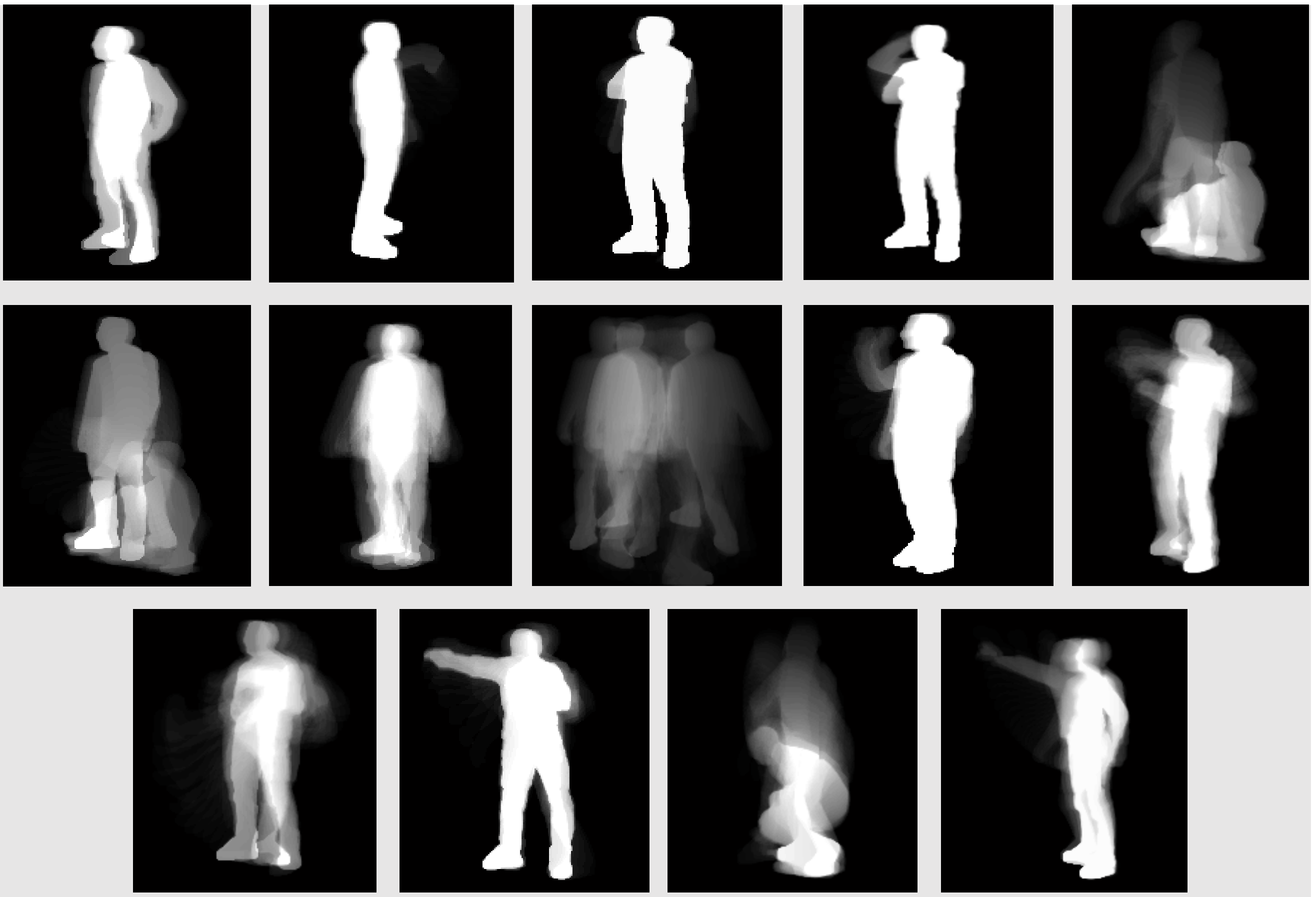


Figure 3.10: AEI image of different action classes of IXMAS datasets: (left to right) 1. Nothing 2. Check watch 3. Cross arm 4. Scratch head 5. Sit down 6. Get up 7. Turn around 8. Walk 9. Wave 10. Punch 11. Kick 12. Point 13. Pick up 14. throw

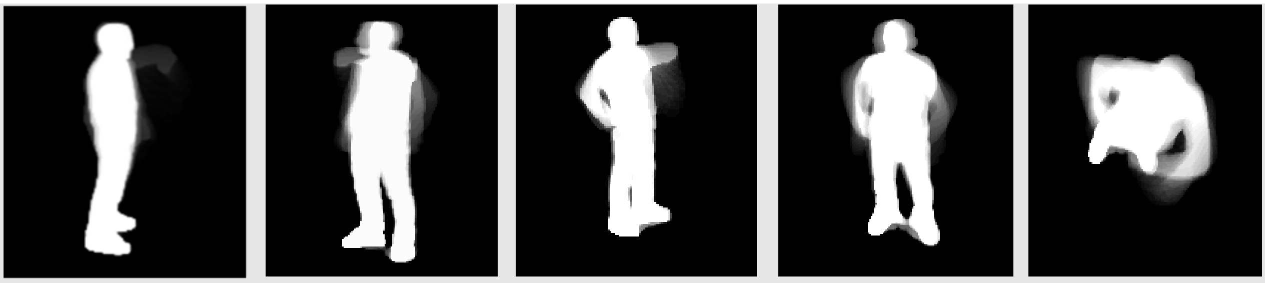


Figure 3.11: AEI image of check watch action class of IXMAS datasets at different camera angles

3.3 SDGs computation

In this section, an image is represented by its distribution among edge orientations i.e. local shape and its spatial arrangement which is dividing the image into blocks at multiple resolutions. SDGs descriptor is a spatial pyramid representation of HOG descriptor The SDGs descriptor is evaluated by computing histogram of orientation gradients in each image sub-region at multiple resolution level and past study shows that it reached good performance. The quantization of edge orientations into K bins over an image sub-region describes the local shape. Each of the bin represents the number of edges having a particular range of angular orientation. The weight of each edge is calculated by its magnitude and contributes accordingly.

The image is divided at several pyramid level to form cells. Along each dimension, resolution level l grid has 2l cells The HOG descriptors are computed for each grid cell at each pyramid level and are concatenated to give SDGs image descriptor [39] [40]. Accordingly, 0 level is depicted by K vector in analogous to the K-bins of the histogram, level 1 represented by a 4K-bin histogram, etc, and for the complete image, SDGs descriptor is a vector with dimensionality $K \sum_{l \epsilon L} 4l$. For example, L= 1 level and K= 8 bins has 40 dimension SDGs-vector. In our method, the levels are constrained up to L=3. To assure that images having more edges are not weighted strongly than others, SDGs is normalized to unity. The vectors at each level are concatenated such that the loss of spatial information is minimized. Figure 3.12 shows the SDGs descriptor at each level.

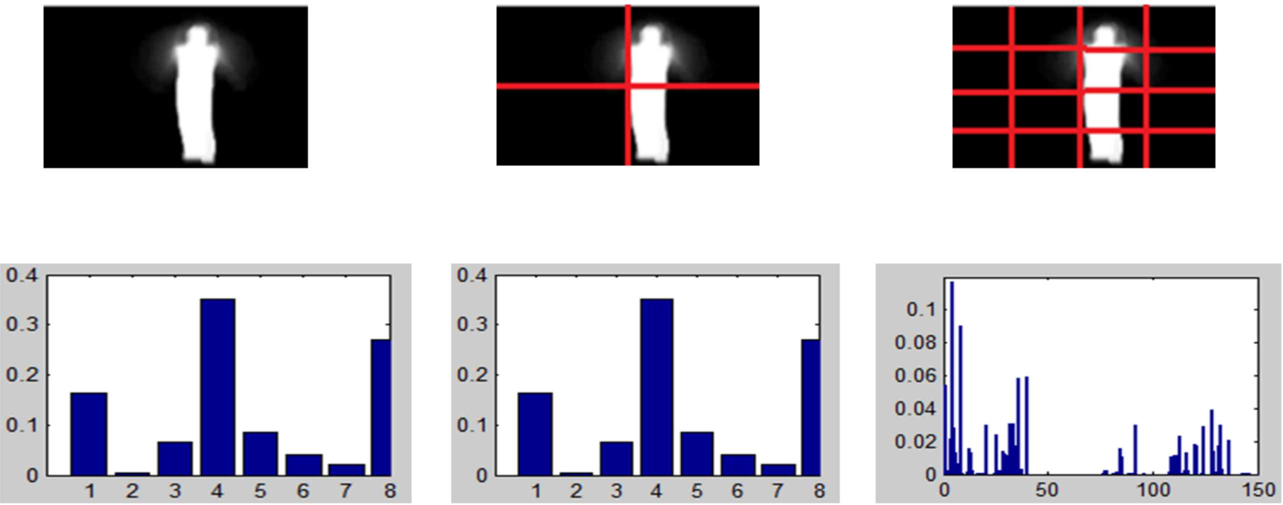


Figure 3.12: PHOG descriptor. (a): Average energy image at level 0. (b), (c): gradient image, level 1 and level 2. (d) (e), (f) : PHOG vector at level 0, level 1 and level 2 accordingly.

In figure 3.12(a), the average energy image is shown which are computed for each and every video of an action class. The (b) part represents the gradient image at level 1. For calculating this image is distributed in 8 bins pyramidically. Then the image is again represented in level 2 gradient for which in which for each gradient part, the histogram is computed by distributing the bins in pyramid shape. The part d, e, f of figure 3.12 represents the corresponding histograms for level 0, level 1 and level 2.

The problem with this method is that if levels are increased then the dimensionality of the feature vector increased drastically which is not required in any HAR system. To remove this problem, the histograms from each level up to 2 are concatenated to form a single vector. This removes the dimensionality problem as well as increases the uniqueness of the feature vector.

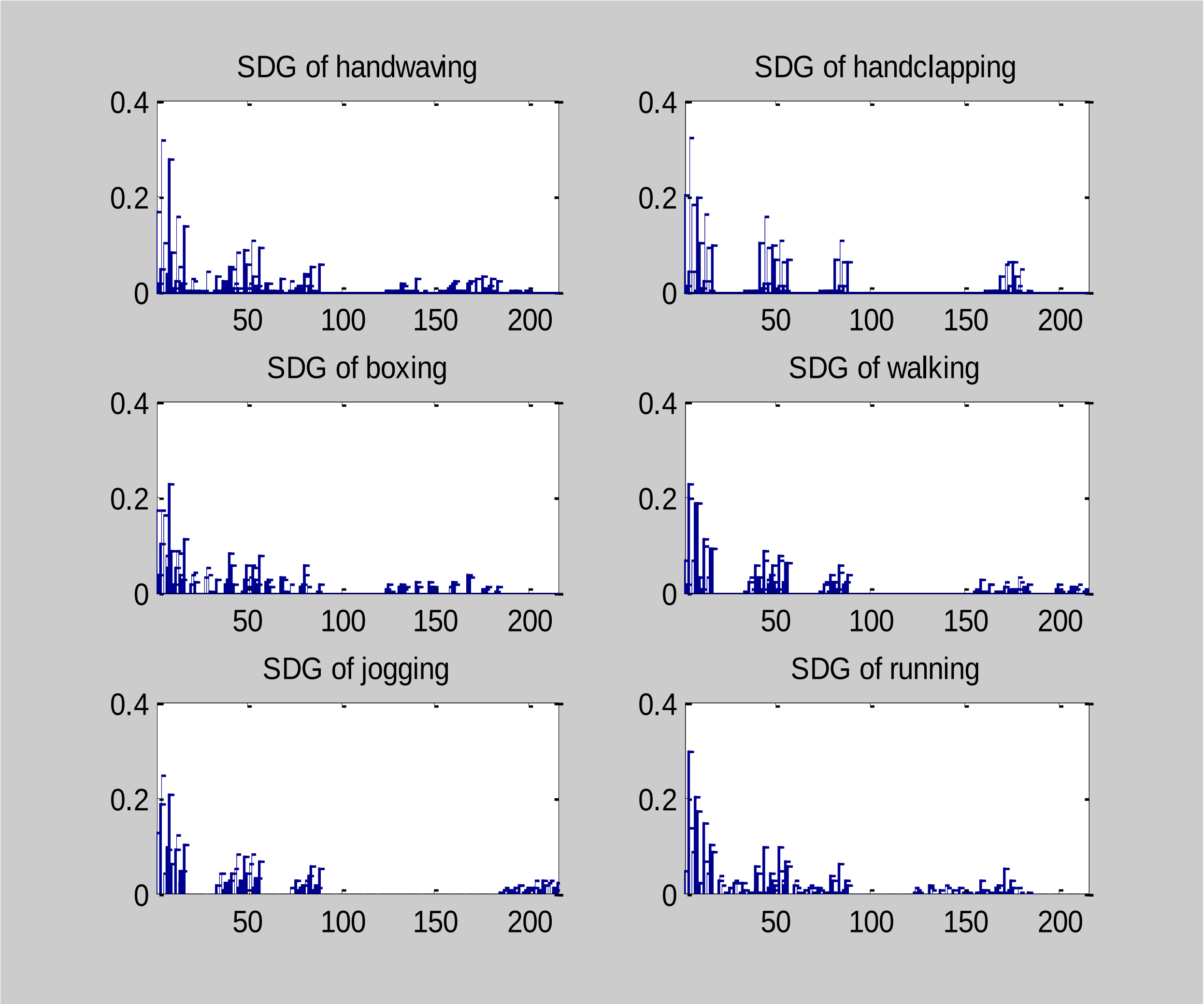


Fig 3.13: SDGs of various action classes of KTH dataset

Figure 3.13 shows the spatio distribution gradients for different activity classes of KTH dataset. There are six different activities. The figure depicts the uniqueness of each activity. it can be seen that each SDG is different from one another. Hence it is regarded as a good feature vector having spatial features. The temporal information is also preserved in this method as the time for which one activity is performed is taken into consideration when calculating the average energy image.

3.5 Computation of Spatio Temporal Interest Keypoints

This section describes in detail some of the more notable local feature detectors. A feature detector finds the points in the video where features are going to be extracted. These points are known as Spatio-Temporal Interest Points (STIPs). A STIP is a point in space and time (*x*,*y*,*t*) that has high saliency. High saliency means that there are high amounts of changes in the neighbourhood of the point. In the spatial domain this shows as large contrast changes, yielding a Spatial Interest Point (SIP). Saliency in the temporal domain occurs when a point changes over time, and when this change occurs at a SIP the point is then a STIP.

The motivation for locating areas in the video having high saliency is that they must be the important areas for describing the action in the scene. This can be confirmed by observing a video of a person running. The difference in appearance between the person and the background will result in high spatial saliency all around the contour of the person.

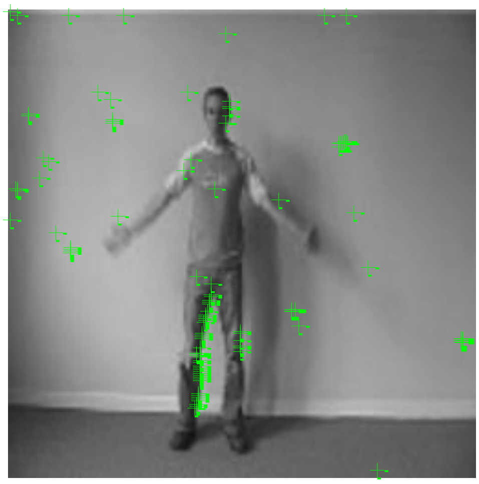


Fig 3.14: Interest points shown in Handwaving activity

The fact that the person is running results in high temporal saliency at the same points, thus giving rise to STIPs. Figure 3.14 shows the spatio temporal interest points during the hand-waving activity. This figure depicts that it contains interest points on the actor as well as on the background. The points on background are considered as invaluable key points hence they are to be removed for better recognition and less dimensional feature vector. The unworthy keypoints are removed by only considering the human contour as region of interest.

In the human action recognition, the detected STIPs should ideally be located on the actor performing the action that the video represents. The STIPs on a human body will represent action primitives such as: “moving leg” or “lifting arm”, and it is the combinations of the STIPs detected in a video that discriminates one video from another.

To achieve good discrimination between classes, it is therefore desirable to maximize the amount of good STIPs, i. e. STIPs on the actors, versus bad STIPs, i. e. STIPs on the background or on motion that does not represent the action in the video. Figure 3.15 shows a frame of detected STIPs of “person running” from the KTH dataset. The detector has detected three points on the background, which will not add any useful information about the action.

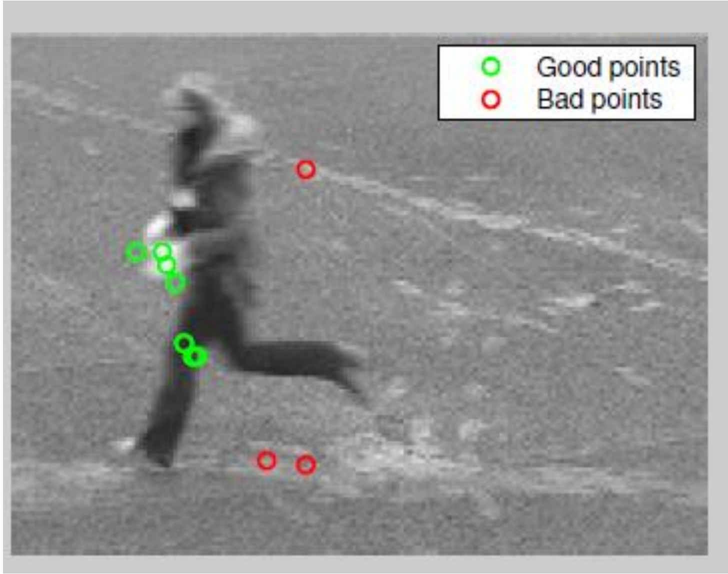


Figure 3.15: Example of good and bad detected feature points. Video is class “running” from KTH dataset.

3.5.1 Scale Space

In the more challenging datasets, there can be large intra-class differences between videos. Videos have different camera viewpoint, scene composition, resolution, etc. This results in that the same object, e. g. a human being, in one video can have the size of hundreds of pixels, while a human in another video only takes up about 50 pixels of space. To be able to detect similar features in these videos, STIPs are detected in different temporal and spatial scales. These scales are represented as a convolution with a Gaussian blurs function, where higher values of the variance σ^2 represent larger scales. This has intuitive meaning because the more the video is blurred the more small details are lost, leaving only larger scale details behind. A video $f(x, y, t)$ is then represented by the scale-space

This stage of the filtering attempts to identify those locations and scales that is identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function. Further it has been shown under reasonable assumptions it must be based on the Gaussian function. The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (3.3)$$

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

Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $D(x, y, \sigma)$ is then given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (3.4)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbours at the same scale, and its 9 neighbours up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

3.5.2 Keypoint Localisation

This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localised on an edge. This is achieved by calculating the Laplacian, value for each keypoint found in stage 1. The location of extremum, **z**, is given by:

$$z = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} \qquad (3.5)$$

If the function value at z is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localisation it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the key point, the key point is rejected.

3.5.3 Orientation Assignment

This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:

- Use the key points scale to select the Gaussian smoothed image L, from above
- Compute:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (3.6)$$

- Compute:

$$\mu(x,y) = \tan^{-1}(L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)) \qquad (3.7)$$

- Form an orientation histogram from gradient orientations of sample points
- Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation
- Some points will be assigned multiple orientations
- Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position

3.5.4 Key point Descriptor

The local gradient data, used above, is also used to create key point descriptors. The gradient information is rotated to line up with the orientation of the key point and then weighted by a Gaussian with variance of 1.5 * key point scale. This data is then used to create a set of histograms over a window centred on the key point.

Key point descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This result in a feature vector containing 128 elements. Figure 3.16 depicts the keypoints in various activities of KTH dataset.

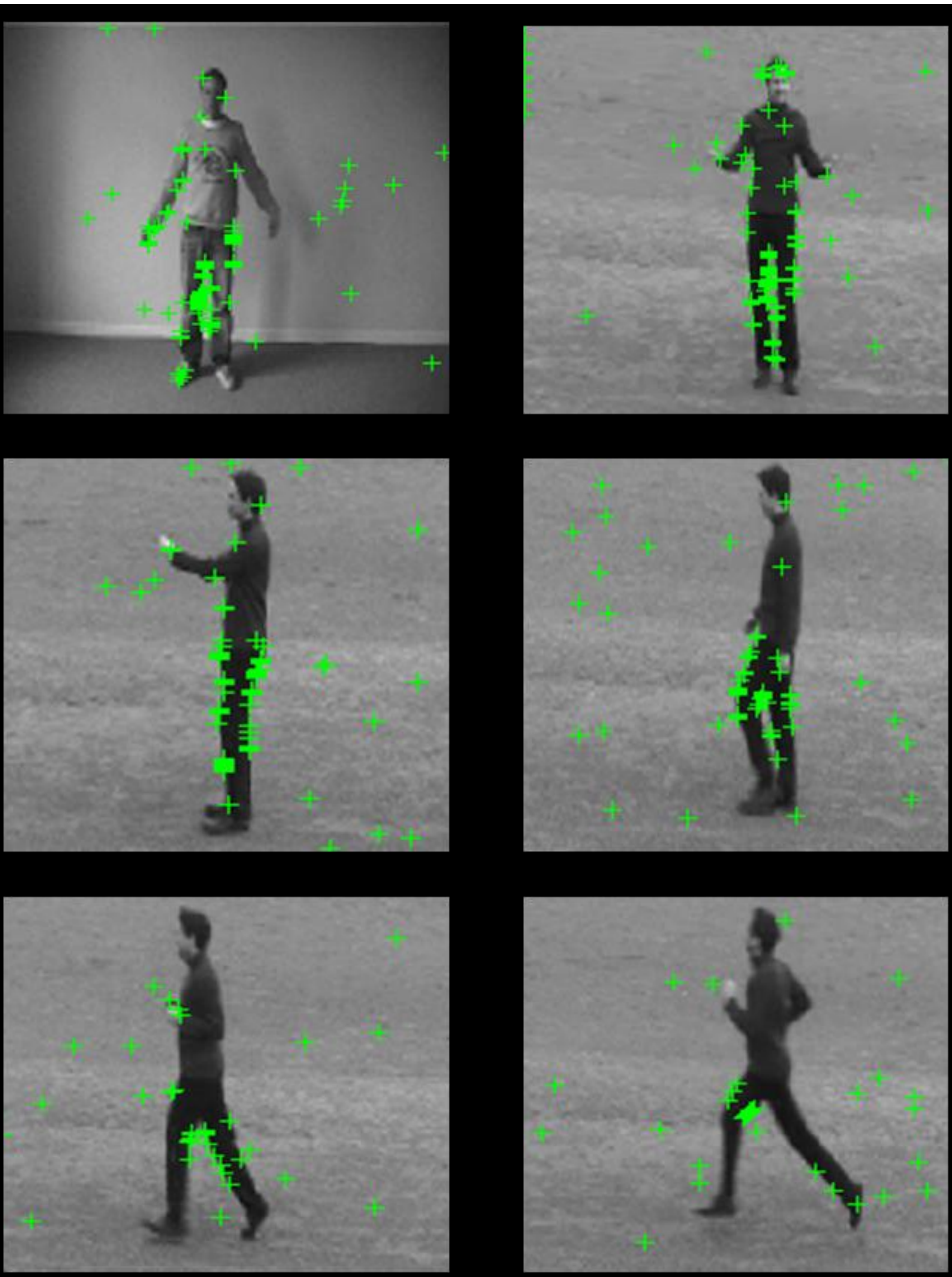


Figure 3.16: Keypoints shown in different activity classes in KTH dataset

Codebook Generation

A vocabulary of spatio temporal interest keypoints is created for better representation. Each frame of the video is represented with interest key points features. There are many numbers of frames in a video. For each frame, there is a keypoint vector which depicts the number of interest points in that frame. Hence for each video of an activity class, keypoint vectors are equivalent in number with the number of frames present in that video. To reduce the amount of data needed for representation, feature vector for various frames are replaced with centroid value of those keypoints vectors. Thus a codebook is created in which one keypoint vector represents one video of an action.

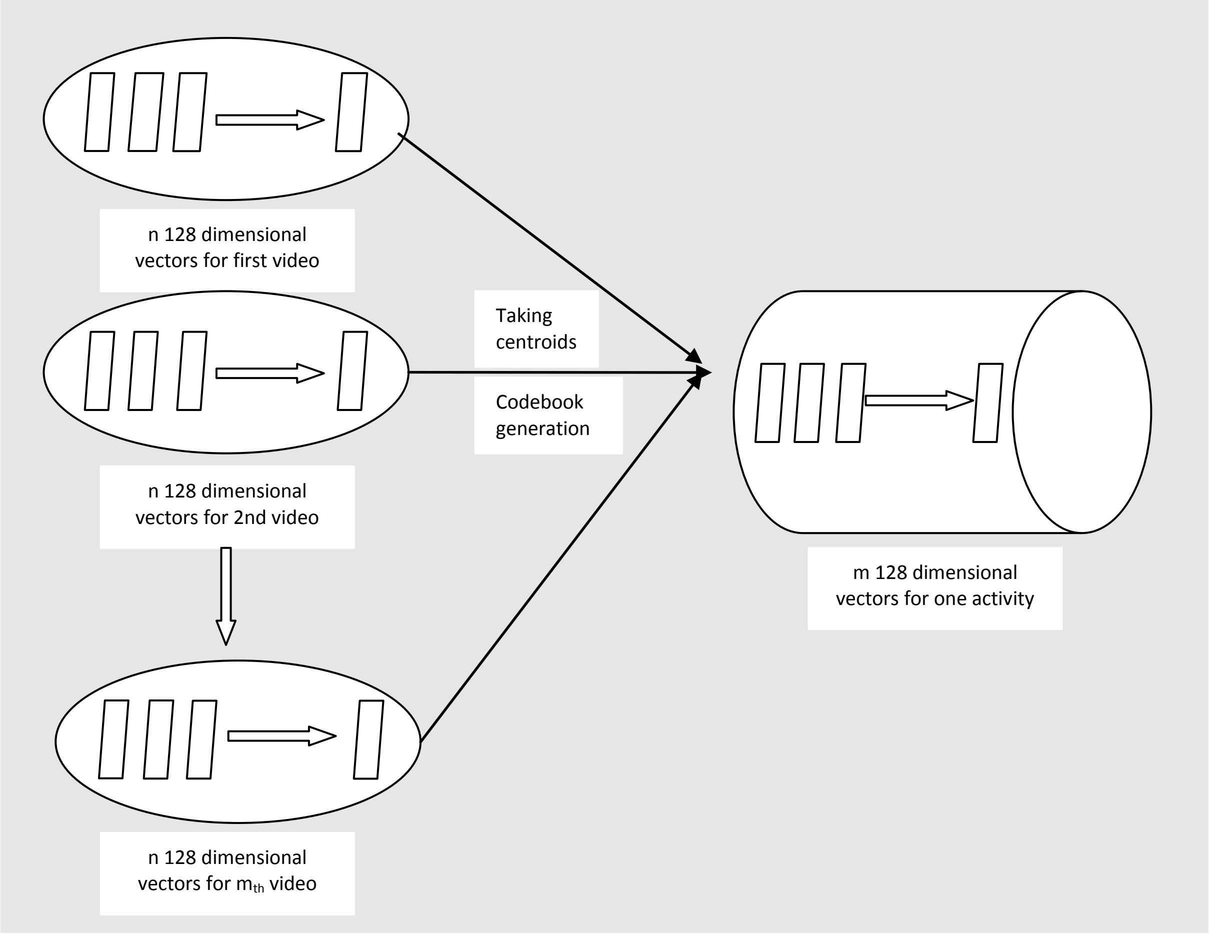

n 128 dimensional vectors for first video
Taking centroids
Codebook generation
n 128 dimensional vectors for 2nd video
m 128 dimensional vectors for one activity
n 128 dimensional vectors for m_{th} video

Figure 3.17: Codebook Generation

3.3 Hybrid Feature Vector

In this work, for each activity video, the video frames are segmented to give silhouettes which are then averaged to evaluate average energy image features. The Region of interest (ROI) containing only the human body is extracted i.e. frame of size 64×48 is used to represent the average energy image of an activity sequence. The Spatial distribution gradient features are evaluated at level 0 to give 8 dimension long vector, level 1 has 40 dimension vector, level 2 has 168 long vector. The concatenated PHOG vector of all three levels has a 216 dimension feature vector. In addition to this, spatio temporal interest points features are also exploited. Key points are of 128 dimensional feature vector which are unique for every activity class. The hybrid feature vector which is used for learning the classifier is of 344 dimensions. The uniqueness of the hybrid feature vector is shown in figure 3.17 which depicts that the each plot values are different among each other.

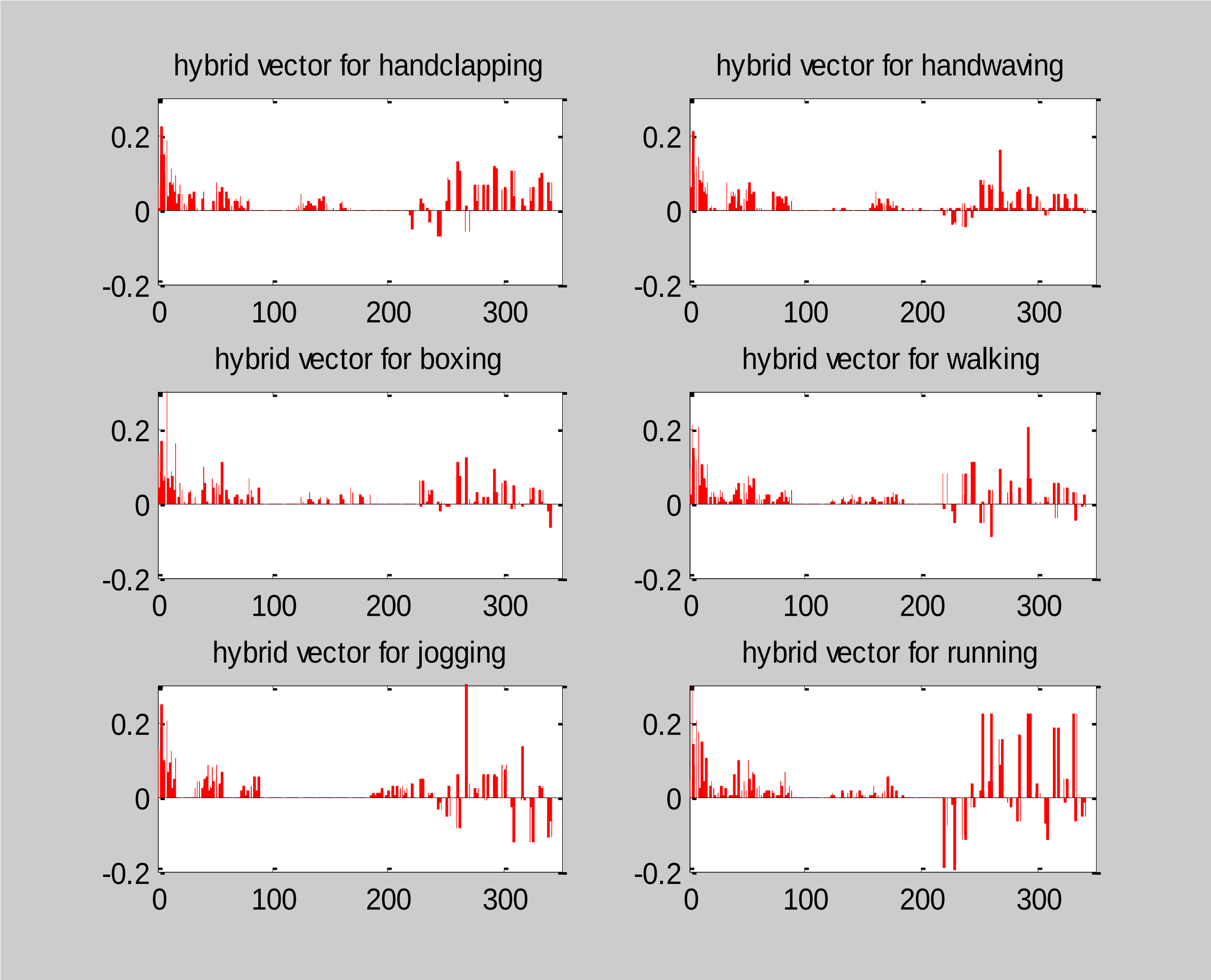


Figure 3.17: Hybrid feature vectors for KTH dataset

For demonstrating the efficacy of the proposed technique, the HAR system is made to implement on four public datasets, the KTH dataset and Weizmann, Ballet and IXMAS dataset. The hybrid feature vector is evaluated for each video of each and every activity classes. For KTH dataset, 60 videos are used for training and 20 videos are used for validation. Similarly for Weizmann dataset, 6 video are used for learning and 3 are used for validation. For Weizmann dataset, LOO strategy is used for classification. The efficiency of the classification is calculated for the datasets and is calculated by equations given below.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FP} \times 100\% \qquad (3.8)$$

Table 3.1 and Table 3.2 represent the classification result in terms of confusion matrix of the proposed technique implemented on KTH and Weizmann datasets with two different classifiers. Table 3.1 show the result for KTH and Table 3.2 represents the results Weizmann dataset. For KTH dataset, 20 sequences are given for testing purpose and the confusion matrix is created. For Weizmann dataset, cross validation is used by adopting LOO (leave one out) strategy by leaving the skip action. Table 3.3 depicts the confusion matrix for ballet dataset by SVM and HMM classifier. The performance of our approach is least on this data set as compared to the Weizmann and KTH data sets, because of the complex motion patterns, which differentiate in execution of the motion from actor to actor. The misclassification error is instigated by the ‘‘hopping’’ as it is confused with a much related activity ‘‘jump’’. Table 3.4 shows the confusion matrix that has been obtained for IXMAS challenging dataset with two Classifiers. As common in the state-of-the-art, we used a leave-one-actor-out cross validation test in which actor invariance is tested by training with the instances from all but one actor and testing the sequences from the unknown one. This is repeated for all available actors and the average accuracy score is obtained. Following the test setup given by the publishers of the dataset, we excluded the point and throw actions

There are various measures of the classification through which the effectiveness of the method is computed. A comparison of our method is made with earlier works. The comparison is made by analyzing the accuracy of classification achieved with different methods. The comparison results shows that our method gives better performance in terms of classification performance parameters. The different classification performance parameters achieved using the proposed method is shown in Table 3.5. The results depicts that our methods gives satisfactory result as compared to other state of the methods.

Confusion Matrix Results for KTH Dataset

Table 3.1(a): Confusion matrix for KTH dataset by using SVM classifier

	Hand Clapping	Hand Waving	Boxing	Walking	Jogging	Running
Hand Clapping	20/20					
Hand Waving		20/20				
Boxing		1/20	19/20			
Walking				20/20		
Jogging				1/20	19/20	
Running					2/20	18/20

Table 3.1(b): Confusion matrix for KTH dataset by using HMM classifier

	Hand Clapping	Hand Waving	Boxing	Walking	Jogging	Running
Hand Clapping	20/20					
Hand Waving		20/20				
Boxing		2/20	18/20			
Walking				20/20		
Jogging				1/20	19/20	
Running					2/20	18/20

Confusion Matrix Results for Weizmann Dataset

Table 3.2(a): Confusion matrix for Weizmann dataset by using SVM classifier

	Bend	Jack	Jump	Pjump	Run	Side	Walk	Wave1	Wave2
Bend	9/9								
Jack		9/9							
Jump			9/9						
Pjump				9/9					
Run					8/9		1/9		
Side						9/9			
Walk					1/9		8/9		
Wave1								9/9	
Wave2									9/9

Table 3.2(b): Confusion matrix for Weizmann dataset by using HMM classifier

	Bend	Jack	Jump	Pjump	Run	Side	Walk	Wave1	Wave2
Bend	9/9								
Jack		9/9							
Jump			9/9						
Pjump				9/9					
Run					8/9		1/9		
Side						9/9			
Walk					1/9		8/9		
Wave1								9/9	
Wave2								1/9	8/9

Confusion Matrix for Ballet Movement Dataset

Table 3.3(a): Confusion matrix for Ballet dataset by using SVM classifier

	Hopping	Jump	LR hand opening	Leg swinging	RL hand opening	Stand still	Turning right
Hopping	16/20	4/20					
Jump	18/20	2/20					
LR hand opening			20/20				
Leg Swinging				20/20			
RL hand opening				1/20	19/20		
Stand still						20/20	
Turning right							20/20

Table 3.3(b): Confusion matrix for Ballet dataset by using HMM classifier

	Hopping	Jump	LR hand opening	Leg swinging	RL hand opening	Stand still	Turning right
Hopping	15/20	5/20					
Jump	2/20	18/20					
LR hand opening			20/20				
Leg Swinging	1/20			19/20			
RL hand opening				1/20	19/20		
Stand still						20/20	
Turning right							20/20

Confusion Matrix for IXMAS dataset

Table 3.4(a): Confusion matrix for IXMAS dataset by using SVM classifier

	Check watch	Cross arm	Scratch head	Sit down	Get up	Turn around	walk	wave	punch	kick	Pick up
Check watch	29/36	5/36							2/36		
Cross arm		33/36	1/36						2/36		
Scratch head	1/36	4/36	25/36						2/36	1/36	3/36
Sit down				34/36							2/36
Get up					36/36						
Turn around						36/36					
Walk						5/36	31/36				
Wave	4/36		6/36					25/36	1/36		
Punch	2/36	1/36	2/36					1/36	30/36		
Kick		2/36							4/36	30/36	
Pick up											36/36

Table 3.4(b): Confusion matrix for IXMAS dataset by using HMM classifier

	Check watch	Cross arm	Scratch head	Sit down	Get up	Turn around	walk	wave	punch	kick	Pick up
Check watch	29/36	5/36							2/36		
Cross arm		33/36	1/36						2/36		
Scratch head	1/36	7/36	20/36						2/36	1/36	3/36
Sit down				34/36							2/36
Get up					36/36						
Turn around						36/36					
Walk						5/36	31/36				
Wave	4/36		6/36					25/36	1/36		
Punch	2/36	1/36	2/36					1/36	30/36		
Kick		2/36							4/36	30/36	
Pick up				2/36							34/36

Table 3.5 shows the classification parameter results obtained by using our hybrid feature used for human activity recognition using SVM and HMM classification model respectively. The results are represented in terms of classification parameters discussed earlier. The values for accuracy are depicted for all the datasets. The values show that the proposed methodology gives suitable performance with reduced length of feature vector.

Table 3.5: Classification accuracy for all
the datasets using SVM and HMM classifier.

Classifier / Dataset	ARA with SVM	ARA with HMM
KTH	96.6	97.5
Weizmann	93.75	98.4
Ballet	100	98.4
IXMAS	96.77	98.4

The most astounding accuracy accomplished on these datasets is contrasted and compared with other methods. Table 3.6 introduces the comparison on Weizmann dataset and Table 3.7 exhibits the comparison on KTH human activity datasets

Table 3.6: Comparison of recognition accuracy with similar state-of-the-art techniques on Weizmann Dataset

Method	**Parameters**		
	Classifiers	***Test scheme***	***ARA (%)***
Gorelick et al. [9]	KNN	LOSO	97.5
Chaaraoui et al. [41]	KNN	LOSO	92.8
Wu, & Shao [32]	SVM	LOSO	97.78
Goudelis et al. [42]	SVM	LOPO	95.42
Touati and Mignotte [43]	KNN	LOO	92.3
Proposed method	SVM , HMM	LOO	97.5, 96.2

Table 3.7: Comparison of recognition accuracy with similar state-of-the-art techniques on KTH Dataset

Method	**Parameters**		
	Classifiers	***Test scheme***	***ARA (%)***
Sadek et al. [44]	SVM	-	93.30
Saghafi and Rajan [45]	KNN	LOO	92.6
Goudelis et al. [42]	SVM	LOPO	93.14
Rahman et al. [46]	KNN	LOO	94.49
Conde and Olivieri [47]	KNN	-	91.3
Proposed method	SVM, HMM	-	96.6, 95.8

The detail of classification result of different activities of Ballet dataset is presented in Table 3.8. The highest ARR achieved is 94.5%, which is less than the earlier datasets used in this work. The key reason for the decrease in the accuracy is the complex motion pattern, which differentiates due to execution of the motion from actor to actor and high misclassification error is introduced due to the "hopping" movement as it is confused with a much-related movement "jumping". It is also observed that the formation of the AESI is difficult because of the complexity. Table 3.9 shows a comparison on IXMAS datasets for different human action recognition approaches. The number of action classes, actors and views has been detailed because these vary among the approaches. Wu et al. (2011) obtained their highest rate excluding camera 4, whereas Cherla et al. (2008) excluded the top-view camera and reorganised the 4 side views into 6 viewing angles in order to achieve view consistency. The test strategies utilized as a part of these methodologies are LOO (Leave One Out), LOSO (Leave-One-Sequence-Out). LOPO (Leave-One-Person-out) and classifier is SVM.

Therefore, the comparison is sufficiently reasonable as the exploratory setup utilized as a part of our methodology is similar to that in other techniques. The high accuracy is accomplished because of the utilization of spatial and temporal data exploited utilizing SDGs-spatio temporal interest key points based feature descriptor. Consequently, it can be said that the proposed system is robust to handle the perspective issues.

Table 3.8: Comparison with other human action recognition approaches of the state of-the art. The accuracy obtained in the leave-one-actor-out cross validation performed on the Ballet dataset

Method	**Parameters**		
	Classifiers	***Test scheme***	***ARA (%)***
Fathi & Mori [33]	*Adaboost*	*LOOCV*	*51*
Wang & Mori [60]	*S-CTM*	*LOO*	*91.3*
Guha & Ward [61]	*RSR*	*LOO*	*91.1*
Ming et al. [62]	*RVM*	*LOO*	*90.8*
Iosifidis et al. [63]	*SVM*	*LOO*	*91.1*
Vishwakarma & Kapoor [45]	*SVM-NN*	*LOOCV*	*94.0*
Proposed Method	*SVM, HMM*	*LOOCV*	*95.62, 94.3*

Table 3.9: Comparison with other multi-view human action recognition approaches of the state of-the art. The accuracy obtained in the leave-one-actor-out cross validation performed on the IXMAS dataset

Method	***Actions***	***Actors***	***Views***	***Accuracy***
Wu et al. (2011)	*Images*	*12*	4	*89.4*
Weinland et al. (2006)	*Silhouettes*	*11*	*5*	*93.3*
Cherla et al. (2008)	*Silhouettes*	*13*	*4*	*80.1*
Weinland et al. (2010)	*Images*	*11*	*5*	*83.5*
Proposed Method	*Images*	*11*	5	*87.12, 86.36*

Chapter 4

Conclusion and Future Scope

In this paper, a human action recognition approach using shape and motion features of the human silhouette in the video sequence is presented, which addresses the problem of less recognition rate under challenging environmental conditions and complex motion pattern. The shape information is computed through modified SDGs of AEI. The spatial and temporal features are exploited through spatio temporal interest key points. As the increase in the decomposition levels of SDGs feature vector, the quality and effectiveness of spatial distribution is less, while the complexity increases exponentially. The STIP computed feature vector results in high dimension and hence, a dimension reduction technique is applied for the compact representation and improved classification using vector quantization.

The approach is based on the human silhouette. Hence, there is a strong need for effective segmentation technique to extract the foreground object as it is very difficult to recognize actions of multiple persons in a single video. The parameters like the number of levels of SDGs computation and the number key frames used for motion estimation may be optimized for higher recognition accuracy with minimal computation.

In this work, a vision based human activity recognition system exploiting the key points is presented. The problem of less recognition rate under the variant environmental conditions has been addressed by employing: (1) Accurate human silhouette extraction through texture based background subtraction approach (2) Simple and effective representation of human silhouettes by average energy image. (3) Feature description using STIP and SDGS. (4) An efficient classification using SVM and HMM. The effectiveness of the proposed approach is tested on three public data sets through SVM and HMM classification models and ARA of these models are measured. The success of these classification models is assessed using ARA and it is observed that the SVM classification model gives better performance than HMM classification model. The overall performance of the proposed approach is found to be comparatively more effective. The parameters used for feature representation are simple and the computation is easy. The four data sets used here vary in terms of lighting conditions, camera positions, indoor and outdoor environment, zoom in, zoom out, and hence it can be concluded that the proposed approach is robust under such conditions.

Despite the satisfactory results, some concerns have cropped up: (i) It is imperative that only one person is in the video sequence, (ii) Some parameters like the number of key points, levels of SDGs can be further optimized (iii) This approach is less effective, when object is occluded.

For future work, a more robust technique can be developed for the HAR, which does not require the segmentation of object. A still image based action recognition may lead to a new destination of HAR, which would require only visual information to recognize the human action.

There are some practical applications, where this approach can be usefull in developing an intelligent system, which can perform the subsequent tasks: (i) To provide assistance to perform the perfect physical exercise, (ii) Gait analysis, which can open the dam for biometric identification, and study of Gait disorder and (iii) To detect the abnormal activity for the elderly monitoring system.

For the future, one can optimize these parameters further so that more effective and accurate representation is conceivable. The same approach may be extended for other avenues of research like Human Style Recognition, Hand Gesture Recognition, Facial Recognition etc. An expert and intelligent system can be developed using this approach for variety of applications such as: (i) Telemedicine system for providing assistance to Parkinson disease patients (ii) An intelligent system, which can monitor the elderly person for abnormal activity, and (iii) Intelligent surveillance system, which can raise an alarm during theft, robbery etc. (iv) A system which can coach athletes to improve their techniques by providing correct assistance e.g. golf swing, cricket swing etc.

Our Action Recognition approach assumes that there are no occlusions and only one subject is moving i.e. single agent is present. Our approach can be extended to multiple agents by having each agent tracked individually and treating one agent at a time. Performance of our method in presence of partial occlusions in few frames should be studied. Our approach could be made robust to view angle changes by incorporating more samples of activities shot from different camera view angles in the training database. More challenging dress problems like fully occluded legs can be handled by obtaining the internal contours from the silhouettes. Also, the Periodicity Template, which we used for periodicity analysis of activities, can also be used for activity representation.

Yet another interesting question is: which aspect should be handled in which part of the system to obtain optimum performance. For example, view invariance can be handled at feature extraction level by finding view invariant features, at training level by incorporating action training samples from different views in the dataset or at the classifier level by using view invariant matching techniques. Thus, in which block of the system the complexity should be introduced such that the overall performance of the integrated system becomes the best is an interesting subject matter for investigation. Robustness, accuracy and speed are the three typical conflicting characteristics of a computer vision system. The order of preference that should be given to these three performance measures depends on the application area. Finally, whether it is possible to build a generic Action Recognition system which will work satisfactorily for different applications is the thought provoking question to answer which we still need an extensive amount of exploring and research in this area.

References

[1] J. K. Agrawal and M. S. Ryoo, *Human activity analysis: A review*, 2011.

[2] D. K. Vishwakarma and R. Kapoor, "Simple and intelligent system to recognize the expression of speech disabled person," in *4th IEEE international conference on intelligent Human Computer Interaction*, Kharagpur, India, 2012.

[3] N. Gkalelis, A. Tefas and I. Pitas, "Human identification from human movements," in *ICIP*, 2009.

[4] A. Iosifidis, A. Tefas and I. Pitas, "Activity based person identification using fuzzy representation and discriminant learning," *TIFS,* vol. 7, no. 2, p. 530–542, 2012.

[5] S. Brutzer, B. Hoferlin and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2011.

[6] D. K. Vishwakarma, R. Kapoor and A. Dhiman, "A unified framework for human activity recognition: An approach using spatial edge distribution and R transform," *International Journal of Electronics and Communication,* vol. 70, pp. 341-353, 2016.

[7] D. K. Vishwakarma, R. Kapoor and A. Dhiman, "A proposed unified framework for the recognition of human activity by exploiting the characteristics of action dynamics," *Robotics and Autonomous Systems,* vol. 77, pp. 25-38, 2015.

[8] C. Schuldt, I. Laptev and B. Caputo, "Recognizing human actions: A local SVM approach," in *International conference on pattern recognition*, 2004.

[9] L. Gorelick, M. Blank, E. Shechtman and M. Irani, "Actions as space–time shapes," *pattern analysis and machine intelligence,* vol. 29, no. 12, p. 2247–2253, 2007.

[10] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," 2008.

[11] D. Weinland, R. Ronfard and E. Boyer, "Free viewpoint action recognition using motion history volumes.," *Computer Vision Image Understanding,* p. 249–257, 2006.

[12] A. F. Bobick and A. D. Wilson, "A state based approach to the representation and recognition of gestures.," *IEEE trans pattern analysis and machine intelligence,* vol. 19, pp. 1325-1337, 1997.

[13] B. Chakraborty, O. Rudovic and J. Gonzalez, "View-invariant human body detection and extension to human action recognition using component wise HMM of body parts," in *IEEE international conference Automatic Face Gesture Recognition.*, 2008.

[14] C. Rao and M. Shah, "View-invariance in action recognition," *IEEE Comput. Sci. Conf. Comput. Vis. Pattern Recog,* vol. 2, pp. 316-322, 2001.

[15] L. Liu, L. Shao, X. Zhen and X. Li, "Learning discriminative key poses for action recognition," *IEEE Trans Cybern,* vol. 43, pp. 1860-1870, 2013.

[16] A. Gilbert, J. Illingworth and R. Bowden, "Action recognition using mined hierarchical computed features," *IEEE trans pattern recognition and machine inteligence,* vol. 33, pp. 883-897, 2011.

[17] N. Ikizler-Cinbis and S. Scarloff, "Web based classifier for human action recognition," *IEEE Trans. Multimedia,* vol. 14, pp. 1031-1045, 2012.

[18] Y. Ke, R. Sukthankar and M. Hebert, "Spatio temporal shape and flow correlation for action recognition," 2007.

[19] J. Liu, J. Luo and M. Shah, "Action recognition in unconstrained amateur videos," in *IEEE Int. Conf. Acoust., Speech Signal Process.,* 2009.

[20] H. Ragheb, S. Velastin, P. Remagnino and T. Ellis, "Human action recognition using robust power spectrum features," in *IEEE Conf Image Process,* 2008.

[21] A. A. Chaaraoui, P. Climent-Pérez and F. Flórez-Revuelt, "A review on vision techniques applied to human behaviour analysis for ambient-assisted living.," *Exp. Systems Appl2012,* p. 10873–10888..

[22] S. Vishwakarma and A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance.," *Visual Computer,* p. 983–1009, 2012.

[23] D. Weinland, R. Ronfard and E. Boyer, "A survey of vision-based methods for action representation, segmentation, and recognition," *Computer Vision and Image Understanding,* vol. 115, p. 224–241, 2011.

[24] M. Ziaeefar and R. Bergevin, "Semantic human activity recognition: A literature review," *Pattern Recognition,* 2015.

[25] R. Poppe, "A survey on vision-based human action recognition.," *Image and Vision Computing,* p. 976–990, 2010.

[26] Bobick, F. Aaron, Davis and W. James, "The recognition of human movement using temporal templates.," *IEEE trans. on pattern analysis and machine intelligence,* p. 257–267, 2001.

[27] P. Dollar, V. Rabaud, G. Cottrell and Belongie, "behavior recognition via sparse spatio-temporal features.," in *In Proceedings of 2nd Joint IEEE international workshop on visual surveillance and performance evaluation of tracking and,* 2005.

[28] B. Chakraborty, M. B. Holte, T. B. Moeslund and J. Gonzalez, "Selectice spatio temporal interest points.," *Computer Vision and Image Understanding,* 2012.

[29] I. Everts, J. V. Gemert and T. Gevers, " Evaluation of color spatio-temporal interest points for human action recognition.," *IEEE Transactions on Image Processing,* pp.

1569-1580, 2014.

[30] I. Jargalsaikhan, S. Little, O. Direkoglu and N. E. O’Connor, "Action recognition based on sparse motion trajectories.," in *In Proceedings of IEEE international conference on image processing,* 2013.

[31] D. Weinland, E. Boyer and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *IEEE International Conference on Computer Vison*, 2007.

[32] D. Wu and L. Shao, "Silhouette Analysis-Based Action Recognition Via Exploiting Human Poses," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 23, no. 2, pp. 236-243, 2013.

[33] O. Masoud and N. Papanikolopoulos, "A method for human action recognition.," *Image and Vision Computing,* p. 729–743., 2003.

[34] P. Scovanner, S. Ali and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition.," in *In Proc. 15th Internat. Conf. on Multimedia, ACM, New York*, New York, 2007.

[35] T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 8, p. 1576–1588, 2012.

[36] K. Singh, D. K. Vishwarkarma, G. S. Walia and R. Kapoor, "Contrast enhancement via texture region based histogram equalization," *Journal of m.*

[37] K. Singh, D. K. Vishwakarma, G. S. Walia and R. Kapoor, "Contrast enhancement via texture region based histogram equaliszation," *Journal of Modern Optics,* 2016.

[38] D. K. Vishwarkarma, P. Rawat and R. Kapoo, "Human activity recognition using Gabor transform and Ridgelet transform," *Proceedings of computer science journal,* vol. 57, pp. 630-636, 2015.

[39] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems Man, and Cybernetics,* vol. 6, p. 610–621, 1973.

[40] T. W. Chua, Y. Wang and K. Leman, "Adaptive Texture-Color based background subtraction for video surveillance," in *19TH IEEE International conference on image processing (ICIP),* 2012.

[41] A. Chaaraoui, P. C. Perez and F. Revuelta, "Sihouette-based human action recognition using sequences of key poses," *Pattern Recognition Letters,* vol. 34, pp. 1799-1807, 2013.

[42] G. Goudelis, K. Karpouzis and S. Kollias, "Exploring trace transform for robust human action recognition," *Pattern Recognition,* vol. 46, no. 12, pp. 3238-3248, 2013.

[43] R. Touati and M. Mignotte, "MDS-Based Multi-Axial Dimensionality Reduction Model for Human Action Recognition," in *Proc. of IEEE canadian Conference on Compter and Robot Vision*, 2014.

[44] S. Sadek, A. A. Hamadi, M. Elmezain, B. Michaelis and U. Sayed, "Human Action Recognition via Affine Moment Invariants," in *21st International conference on Pattern Recognition*, 2012.

[45] B. Saghafi and D. Rajan, "Human action recognition using Pose-based disriminant embedding," *Signal Processing: Image Communication,* vol. 27, pp. 96-111, 2012.

[46] S. Rahman, I. Song, M. K. H. Leung and I. Lee, "Fast action recognition using negative space features," *Expert Systems with Applications,* vol. 41, pp. 574-587, 2014.

[47] I. G. Conde and D. N. Olivieri , "A KPCA spatio-temporal differential geometric trajectory cloud classifier for recognizing human actions in a CBVR system," *Expert Systems with Applications,* vol. 42, no. 13, p. 5472–5490, 2015.

[48] D. K. Vishwakarma and R. Kapoor, "Hybrid classifier based human activity recognition using the silhouette and cells," *Expert Systems with Applications,* vol. 42, no. 20, p. 6957–6965, 2015.

[49] J. Han and B. Bhanu, "Individual recognition using gait energy image," *Pattern Analysis and Machine Intelligence, IEEE Transactions,* vol. 28, no. 2, p. 316–322, 2006.

[50] A. Bosch, A. Zisserman and X. Munoz, "Representing shape with a spatial pyramid kernel," in *International Conference on Image and Video Retrieval*, 2007.

[51] B. Zhang, Y. Song and S. U. Guan, "Historic chinese architectures image retrieval by SVM and pyramid histogram of oriented gradients features," *International Journal of Soft Computing,* vol. 5, no. 2.

[52] D. K. Vishwakarma and K. Singh, "Human activity recognition based on spatial distribution of gradients at sub levels of average enrgy silhouette images," *IEEE transactions on Cognitive and Development system,* 2016.

[53] J. C. Niebles, H. Wang and L. Fei-Fei, "Unsupervised learning of humanaction categories using spatio temporal words," *International journal of computer vision,* vol. 79, pp. 299-318, 2008.

[54] G. Lavee, M. Rudzsky and E. Rivlin, "Propagating certainty in Petrinets for activity recognition," *IEEE Trans. Circuit System Video Technology,* vol. 23, pp. 326-337, 2013.

[55] T. Kanade and M. Hebert, "First-person vision," *Proc. IEEE,* vol. 100, p. 2442–2453, 2012.