A COMPARATIVE STUDY ON HUMAN ACTIVITY RECOGNIZATION FROM VIDEO

A thesis submitted in partial fulfillment of the requirement for degree of **Master of Computer Application** in the department of computer science & engineering

Of

Jadavpur University

By

Tanmoy Ghosh

Registration No.: 133670 of 2015-2016

Examination Roll No.: MCA186008

Under the Guidance of

Prof. Debotosh Bhattacharjee

Department of Computer Science & Engineering

Jadavpur University, Kolkata-700032

India

2018

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

To Whom It May Concern

I hereby recommend that the thesis entitled “A comparative study on Human Activity Recognization from video” has been carried out by Tanmoy Ghosh(Reg. No.:-133670 of 2015-2016, Exam Roll), under my guidance and supervision may be accepted in partial fulfillment for the degree of Master of Computer Application in Department of Computer Science and Engineering, Jadavpur University.

……………………………………...................

Prof. Debotosh Bhattacharjee(Thesis Supervisor)
Department of Computer Science and engineering
Jadavpur University, Kolkata-700032

…………………………………………….

Prof. Ujjwal Maulik
Head, Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032

……………………………………………..

Prof. Chiranjib Bhattacharjee
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata-700032

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Certificate of Approval

This is to certified that the thesis entitled “A comparative study on Human Activity Recognization from video” is a bonafide record of work carried out by Tanmoy Ghosh in fulfillment of the requirements for the award of the degree of Master of Computer Application in Computer Technology in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, the opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

……………………………………..

Signature of Examiner 1

Date:

…………………………………….

Signature of Examiner 2

Date:

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Declaration of Originality& Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as a part of his Master of Computer Application studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Tanmoy Ghosh

Registration No.:133670 of 2015-2016

Examination Roll No.:MCA186008

Thesis Title: A comparative study on Human Activity Recognization from video.

……………………………………………

Signature with Date

Acknowledgement

The writing of the thesis, as well as the related work, has been a long journey with input from many individuals, right from the first day till the development of the final project.

With my most sincere and gratitude, I would like to thank **Dr. Debotosh Bhattacharjee**, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestion and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work with him.

I would like to thank all the faculty members of the Department of Computer Science and Engineering of Jadavpur University for their continuous support.

Last, but not the least, I would like to thank all my batch mates of Master of Computer Application in Jadavpur University for staying by side when I need them.

……………………………………….

Name: Tanmoy Ghosh

Registration No.: 133670 of 2015-2016

Examination Roll No.:MCA186008

ABSTRACT

This thesis represents a study on human activity recognization from video. In this work, I proposed an effective way for human activity recognization by detecting the moving part of the human body by using Gaussian Mixture Model and people detection technique. Video file is segmented as frames in the form of RGB images, and these images are used for feature extraction.

Publicly available Weizmann dataset and KTH dataset are used for both training sample and test sample. From each frame of a video moving human part is identified first and the identified part is used for feature extraction. For a moving activity like running, walking, jumping velocity and width is calculated from the centroid. For still activity like handclapping, hand waving frame subtraction technique is used to find the centroid. The experimental results datasets clearly validate the efficiency of the proposed technique.

Keywords: -Foreground Detection, Edge Detection, Centroid, Gaussian Mixture Model, people detection

CONTENTS

CHAPTER 1

INTRODUCTION

Recognizing human actions has become a popular research topic of computer vision. A reliable and effective solution to this problem is essential for a large variety of applications ranging from video surveillance and monitoring to human-computer interaction systems.

There are different ways to represent actions and extract features for action recognition. In some studies, motion-based methods are exploited, whereas actions can also be defined as space-time shapes or space-time interest points for feature extraction. Moreover, in shape and motion-based prototype trees were constructed and in form and motion features were combined for action recognition.

One of the basic ideas in my work is that detection of the foreground object and uses it for feature extraction. There are lots of methods by which we can detect foreground object Cluster-Based Background Modeling Algorithm, Temporal average filter, Gaussian mixture model, Teknomo–Fernandez algorithm, but we use Gaussian mixture model, HOG features and SVM classifier for foreground object detection.

For moving activity (e.g., running, walking, jumping) velocity and body stretchiness is calculated with respect to its position in the frame and for still activity (e.g.- hand waving, hand clapping) frame subtraction method is used to calculated the centroid of the each frame depending on the key pose of the human action.

a) Locate foreground human object: This project totally depends on the accuracy of foreground detection. Gaussian Mixture Model is used for foreground detection of moving activity (e.g. running, walking, jumping) and HOG feature and SVM classifier is used for still activity (e.g. hand waving, hand clapping).

b) Action region selection: Any human action is a combination of movement of one or more regions of the human body. Identification of these active regions and their features are the basis to define any action. Higher accuracy in the selection of active region ensures better efficiency in activity recognization.

c) The Classifier: It ensures the efficiency to recognize the actual action.

There are still many issues and challenges that motivate the development of new activity recognition techniques to improve the accuracy under more realistic conditions. Challenges corresponding with activity recognition have been discussed in researchers[1-5, 18]. A number of these challenges are:

- *Human behavior*: performing multiple tasks at the same time makes the recognition process more difficult [3–5].
- *The definition of physical activities*: develop a clear understanding of the definition of the activities under investigation and their specific characteristics [1].
- *Intraclass variability*: the same activity may be performed differently by different individuals [1].
- *Intraclass similarity*: classes that are fundamentally different, but that show very similar characteristics in the sensor data [1, 5].
- *Selection of attributes and sensors*: the selection of the attributes to be measured and the sensors that measure it plays an important role in recognition performance [1, 2].
- *Sensor inaccuracy*: the sensor data play an important role in the overall recognition results [3].
- *Sensor placement*: the wrong placement or orientation of sensors could be causing a problem or affect the recognition performance [3, 4].
- *Resource constraints*: power consumption is the main factor affecting the size of the battery and sensor nodes (if using inertial sensor) [2–4].
- *Usability*: the systems should be easier to learn and more efficient to use [3].
- *Privacy*: sensitive user information should be not invading users' private life[3].
- *Subject sensitivity*: The accuracy of activity recognition is profoundly affected by the subjects participated in training and testing stages [4].
- *Obtrusiveness*: HAR systems should not require the user to wear many sensors nor interact too often with the application [2][4].
- *Data collection*: collection of training data under realistic conditions [1][2].
- *Flexibility:* the flexibility to support new users without the need of re-training the system [2][4].

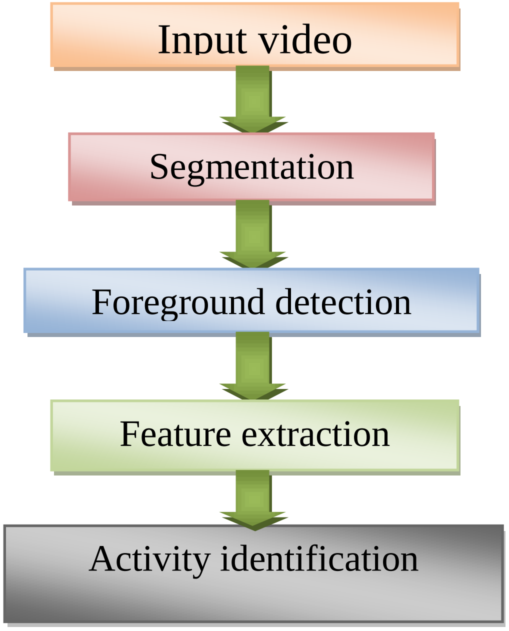
- *Processing:* where the recognition task should be done, whether in the server or the integration device [2].
- *Tradeoffs in HAR*: the tradeoff between accuracy, system latency, and processing power [1].
- *Multiple residents*: More than one resident can be present in the same environment [1].

And of course another challenge is corresponding to the application domain itself, but we present the common and the most popular.

1.1 The objective of This Thesis

The mainobjective of this project is to recognize human activity in a very simple mathematical way from video databases using some simple and effective feature extraction technique. The process includes:

I. Segmenting and processing video frames through resizing.
II. Different foreground detection technique to separate moving and still activity.
III. Extract features to calculate velocity and body stretchiness to separate moving activities.
IV. Extract features to calculate the change of upper body mass to separate still activities.
V. Compute accuracy of the proposed method.



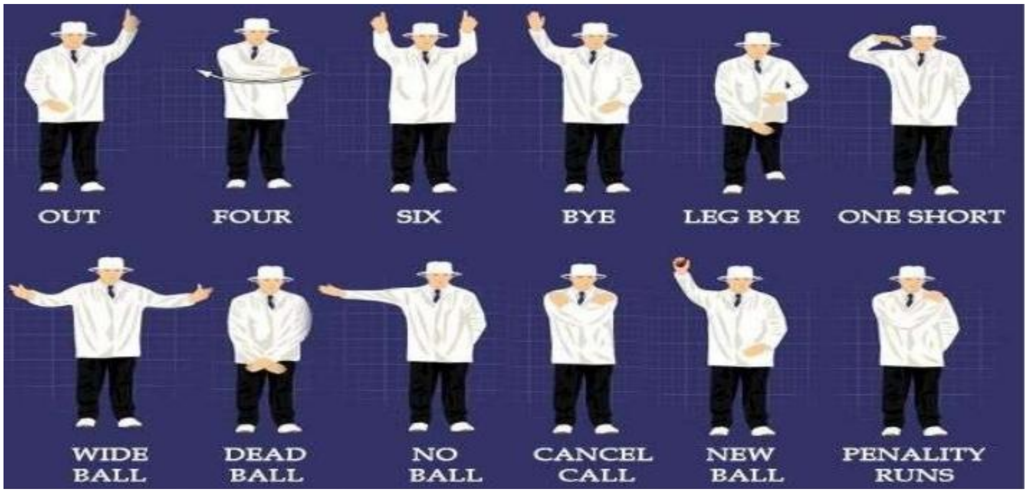
1.2 Levels of human activities[21]

If we categorized human activity based on their complexity:

1) *Actions:* single actor movements,e.g., running, walking, bending, etc.
2) *Interactions:* human-human/ object interactions e.g. punching, lifting bags etc.
3) *Group Activities:* activities of groups, e.g. group dancing, group stealing, etc.

1.3 Applications of human activity recognition

1) *Surveillance:* cameras installed in areas that may need monitoring such as banks, airports, military installations and convenience stores. Currently, surveillance systems are mainly for recording. The aim of activity detection using CCTV's is to monitor suspicious activities for real-time reactions like fighting and stealing.
2) *Sports play analysis:* analyzing the play and deducing the actions in the sports, given below:



3) *Unmanned Aerial Vehicles (UAV's)*: Automated understanding of aerial images. Recognition of military activities like border security, people in bunkers, etc.

1.4 Related Works[18]

Activity recognition became a vital research issue related to the successful realization of intelligent pervasive environments. It is the process by which an actor's behavior and his or her environment are monitored and analyzed to infer the activities. Activity recognition consists of activity modeling, behavior and environment monitoring, data processing and pattern recognition [6]. Activity recognition systems typically have three main components:

- A low-level sensing module that continuously gathers relevant information about activities using microphones, accelerometers, light sensors, and so on

- A feature processing and selection module that processes the raw sensor data into features that help discriminate between activities

- A classification module that uses the features to infer what activity an individual or group of individuals is engaged

- In, for example, walking, cooking, or having a conversation.

There are several approaches for activity recognition as described as follows.

A. **Vision-Based Activity Recognition**

   It uses visual sensing facilities: camera-based surveillance systems to monitor an actor's behavior and the changes in its environment. It is composed of four steps: human detection, behavior tracking, activity recognition and high-level activity evaluation. Various other research approaches used different methods such as a single camera or stereo and infrared to capture activity context. These image-based approaches use single or multiple cameras to reconstruct the 3D human pose, to detect the coordinates of the joints and to extract the limbs of the body. The image analysis is possible by isolating the human body from the background. This is achieved using the background subtraction algorithm that adapts to the environmental changes.

B. **Sensor-Based Activity Recognition**

   It uses sensor network technologies to monitor an actor's behavior along with its environment. In this case, there are sensors attached to humans. Data from the sensors are collected and analyzed using data mining or machine learning algorithms to build activity models and perform activity recognition. In this case, they're recognized activities included human physical movements: walking, running, sitting down/up. Most of the wearable sensors are not very suitable for real applications due to their size or battery life. In sensor-based approach, can use either wearable sensors or object-attached sensors. The most used machine learning is the Hidden Markov Model (HMM) – a graphical oriented method to characterize real-world observations regarding state models. Another good alternative is the Conditional Random Field (CRF) model, which are un-directed graphical methods which allow the dependencies between observations and the use of incomplete information about the probability distribution of a particular observable.

C. Human-Sensing Taxonomy

Classify under the large umbrella of \human-sensing" the process of extracting any information regarding the people in any environment. This describes the inference of spatiotemporal properties (STPs) only. These consist of low-level components regarding the position and history of people in a situation. More specifically:

1) Presence: Is there at least one person present? Presence is arguably the property that is most commonly sought- after in existing real-world applications, the most popular presence-sensor being motion sensors (PIR) and proximity sensors (scalar infrared rangefinders). In cooperative scenarios, though, where people can be instrumented with portable or wearable devices, solutions such as RFID (radio-frequency identification) are becoming increasingly common.
2) Count: How many people are present? The number of people in an environment can be inferred by either employing a person-counting sensor (or sensors) that cover the entire area of interest or by counting people at all the entry and exit points. Commercial people-counting solutions range from thermal imagers [SenSource] and break-beams to simple mechanical barriers such as turnstiles.
3) Location: Where is each person? Location-detection, or \localization", consists of obtaining the spatial coordinates of a person's Centre of mass. Localization can be achieved using instrumented (such as GPS) or fully

un- instrumented solutions (such as cameras). Also, since a grid of presence sensors can also be used to localize people, localization can be considered a higher-resolution generalization of presence detection.

4) Track: Where was this person before? Tracking is the process of solving the correspondence problem that is, extracting the spatiotemporal history of each person in a scene. Equivalently, tracking may be described as recovering a person's relative identity2. For example, if upon detection a person is labeled with a temporary ID (e.g., \person 6") then tracking is the problem of specifying at each subsequent sampling of the scene which detection is the same \person 6". This temporary ID is typically lost in the presence of sensing gaps, such as when the person leaves the scene and returns on the next day. At that point, yesterday's \person 6" will be given a new ID when re-detected. Situations that lead to the loss of a person's relative ID are often called ambiguities. In the remainder of this text, it will use the term piecewise tracking to qualify a tracker that is not capable of adequately handling ambiguities.

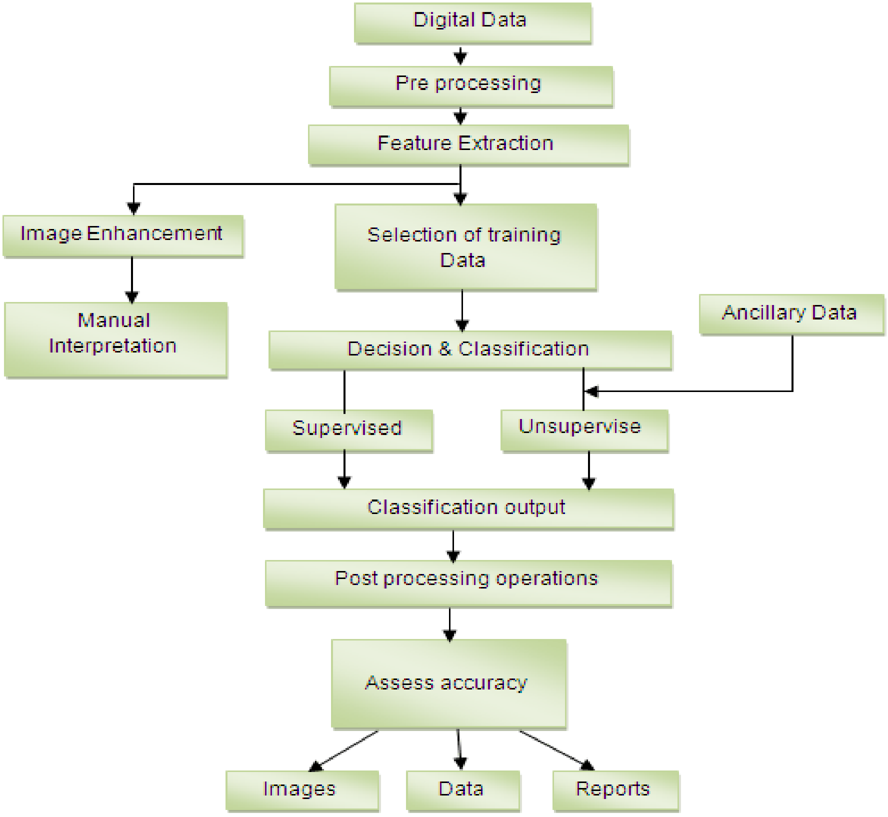
5) Identity: Who is each person? Is this person John? At first glance, it may seem odd to group \identity" into the category of spatiotemporal properties. However, identification is nothing more than a natural extension of tracking where each person is always assigned the same globally unique ID rather than solely relative IDs. Therefore, identity-detection extends tracking so that it becomes possible to recover a person's spatial-temporal history even across sensing gaps.

CHAPTER 2

CONCEPT ON PROJECT BASED TOPICS

2.1 Image Processing

Image processing is a process to perform some operation on an image to extract features associated with that image. Here input is an image and output can be an image or features. The use of image processing technologies is increasing day by day in technologies or research area. For image processing, we use two types of methods – analog image processing and digital image processing. Here we use only digital images, so we consider digital image processing, which techniques help in manipulating the digital images by using computers. Digital image processing technique uses three general phases for processing of all types of data. The phases are pre-processing, enhancement, and display, information extraction.



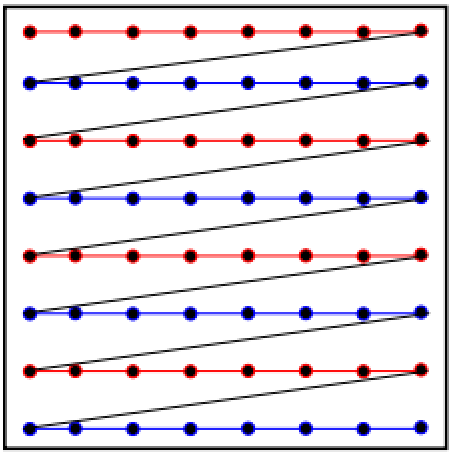
Steps of image processing [24]

2.2 Video Processing [31]

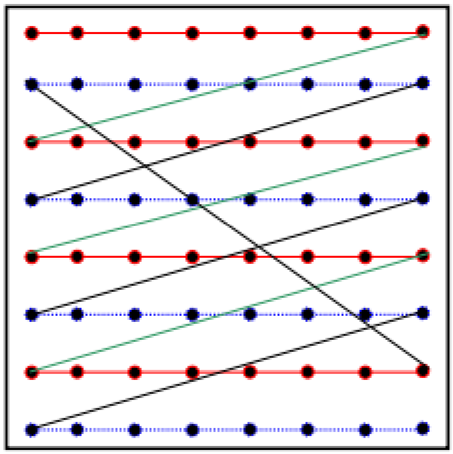
Definition of Video Signal: Video signal is any sequence of time-varying images. A still image is a spatial distribution of intensities that remain constant with time, whereas a time-varyingimage has a spatial intensity distribution that varies with time. Video signal is treated as a series of images called frames. An illusion of continuous video is obtained by changing the frames in a faster manner which is generally termed as frame rate.

Analog Video Signals: Despite the advance of digital video technology, the most common consumer display mechanism for video still uses analog display devices such as CRT. Until all terrestrial and satellite broadcasts become digital, analog video formats will remain significant. The three major Analog Video Signal formats are NTSC (National Television Systems Committee), PAL (Phase Alternate Line) and SECAM (Sequential Color with Memory). All the three are television video formats in which the information in each picture is captured by CCD or CRT is scanned from left to right to create a sequential intensity signal. The formats take advantage of the persistence of human vision by using interlaced scanning pattern in which the odd and even 13 2 lines of each picture are read out in two separate scans of the odd and even fields respectively. This allows good reproduction of movement in the scene at the relatively low field rate of 50 fields/sec for PAL and SECAM and 60 fields/sec for NTSC.

Progressive and Interlaced Scan Pattern: Progressive scan patterns are used for high-resolution displays like computer CRT monitors Digital cinema projections. In progressive scan, each frame of picture information is scanned thoroughly to create the video signal. In interlaced scan pattern, the odd and even lines of each picture are read out in two separate scans of the odd and even fields respectively. This allows good reproduction of movement in the scene at relatively low field rate. The progressive and interlaced scan patterns are shown below.

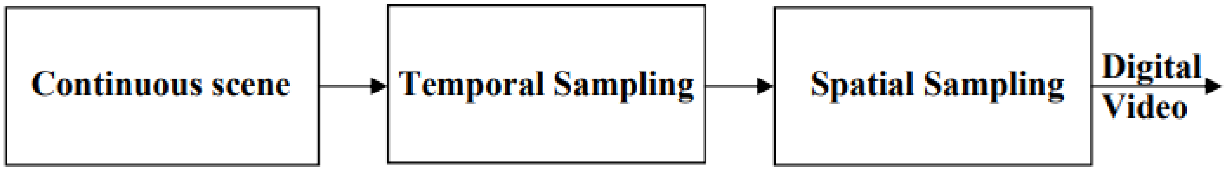


Progressive Scan Pattern



Interlaced Scan pattern

Digital Video: In a digital video, the picture information is digitized both spatially and temporally, and the resultant pixel intensities are quantized. The block diagram depicting the process of obtaining digital video from the continuous natural scene is shown below.



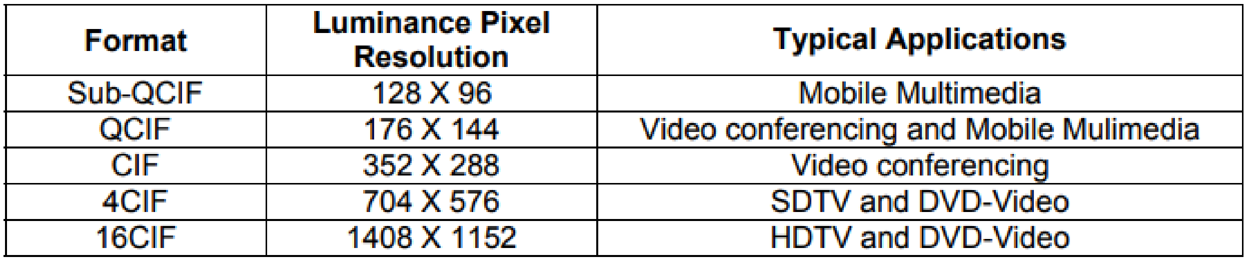
Digital Video from natural scene

The demand for digital video is increasing in areas such as video teleconferencing, multimedia authoring systems, education, and video-on-demand systems.

Spatial Sampling: The sensitivity of Human Visual System (HVS) varies according to the spatial frequency of an image. In the digital representation of the image, the value of each pixel needs to be quantized using some finite precision. In practice, 8 bits are used per luminance sample.

Temporal sampling: A video consists of a sequence of images, displayed in rapid succession, to give an illusion of continuous motion. If the time gap between successive frames is too large, the viewer will observe jerky motion. The sensitivity of HVS drops off significantly at high frame rates. In practice, most video formats use temporal sampling rates of 24 frames per second and above.

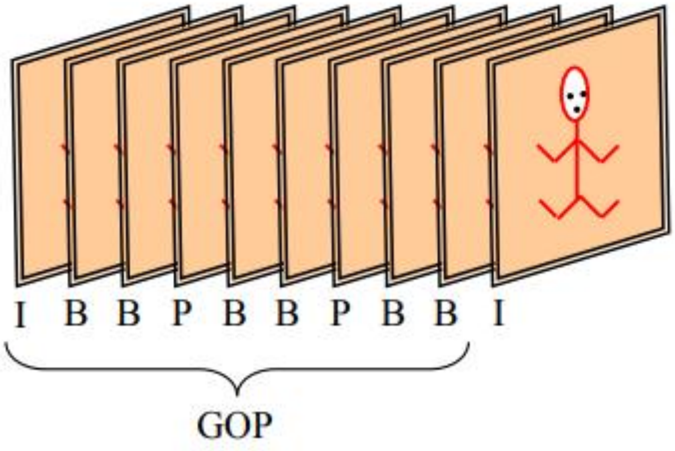
Video formats: Digital video consists of video frames that are displayed at a prescribed frame rate. A frame rate of 30 frames/sec is used in NTSC video. The frame format specifies the size of individual frames in terms of pixels. The Common Intermediate Format (CIF) has 352 x 288 pixels, and the Quarter CIF (QCIF) format has 176 x 144 pixels. Some of the commonly used video formats are given in table 1. Each pixel is represented by three components: the luminance component Y, and the two chrominance components C_b and C_r.



Format	Luminance Pixel Resolution	Typical Applications
Sub-QCIF	128 X 96	Mobile Multimedia
QCIF	176 X 144	Video conferencing and Mobile Mulimedia
CIF	352 X 288	Video conferencing
4CIF	704 X 576	SDTV and DVD-Video
16CIF	1408 X 1152	HDTV and DVD-Video

Video formats

Frame Type: Three types of video frames are I-frame, P-frame and B-frame. ‘I’ stands for Intra coded frame, ‘P’ stands for Predictive frame and ‘B’ stands for Bidirectional predictive frame. ‘I’ frames are encoded without any motion compensation and are used as a reference for future predicted ‘P’ and ‘B’ type frames. ‘I’ frames however require a relatively large number of bits for encoding. ‘P’ frames are encoded using motion compensated prediction from a reference frame which can be either ‘I’ or ‘P’ frame. ‘P’ frames are more efficient in terms of number of bits required compared to ‘I’ frames, but still require more bits than ‘B’ frames. ‘B’ frames require the lowest number of bits compared to both ‘I’ and ‘P’ frames but incur computational complexity. Frames between two successive ‘I’ frames, including the leading ‘I’ frame, are collectively called a group of pictures (GOP). The GOP is illustrated in figure 3. The illustrated figure has one ‘I’ frame, two ‘P’ frames and six ‘B’ frames. Typically, multiple ‘B’ frames are inserted between two consecutive ‘P’ or between ‘I’ and ‘P’ frames. The existence of GOPs facilitates the implementation of features such as random access, fast forward or fast and normal reverse playback shown below.



Group of Picture

Video Processing: Video processing technology has revolutionized the world of multimedia with products such as Digital Versatile Disk (DVD), the Digital Satellite System (DSS), high definition television (HDTV), digital still and video cameras. The different areas of video processing includes (i) Video Compression (ii) Video Indexing (iii) Video Segmentation (iv) Video tracking etc.

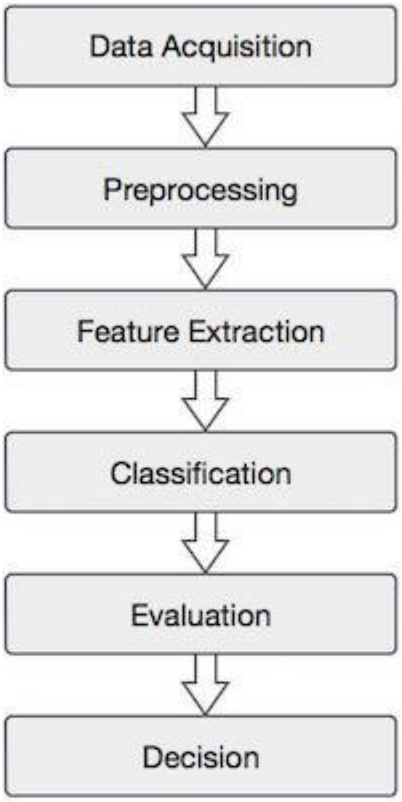
2.3 Pattern Recognition [19]

Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, although it is in some cases considered to be nearly synonymous with machine learning. Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), but when no labeled data are available, other algorithms can be used to discover previously unknown patterns (unsupervised learning).

The terms pattern recognition, machine learning, data mining and knowledge discovery in databases (KDD) are hard to separate, as they largely overlap in their scope. Machine learning is the common term for supervised learning methods and originates from artificial intelligence, whereas KDD and data mining have a larger focus on unsupervised methods and a stronger connection to business use. Pattern recognition has its origins in engineering, and the term is widespread in the context of computer vision: a leading computer vision conference is named Conference on Computer Vision and Pattern Recognition. In pattern recognition, there may be a higher interest to formalize, explain and visualize the pattern, while machine learning traditionally focuses on maximizing the recognition rates. All of these domains have evolved substantially from their roots in artificial intelligence, engineering, and statistics, and they've become increasingly similar by integrating developments and ideas from each other.

In machine learning, pattern recognition is the assignment of a label to a given input value. In statistics, discriminant analysis was introduced for this same purpose in 1936. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of *classes* (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation. This is opposed to *pattern matching* algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort of textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern recognition, pattern matching is not generally a type of machine learning, although pattern-matching algorithms (especially with fairly general, carefully tailored patterns) can sometimes succeed in providing a similar-quality output of the sort offered by pattern-recognition algorithms.



Steps of pattern recognition [25]

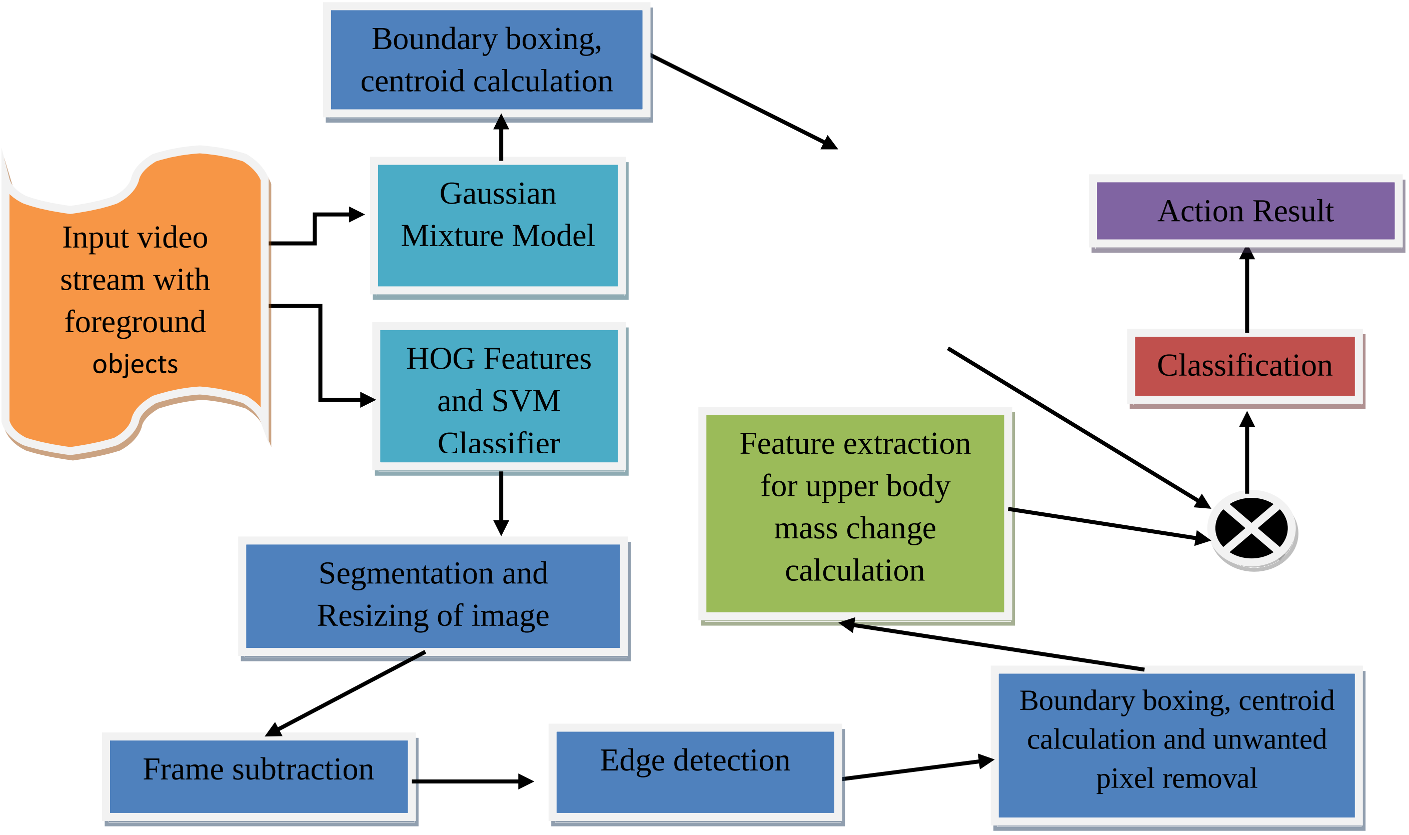
CHAPTER 3

MATERIALS AND METHODS

In thissection a detailed process for single human activity detection using video frames. The proposed method comprises mainly four steps: segmentation, foreground detection, processing, classification.

Methods used in this project are written step by step:

1. In this project, gray images which are extracted from video frames are used. Total around 110 videos are used for each activity detection form KTH and Weizmann data set.
2. Most of these videos are taken using still camera with a still background as well. From each video first frames are extracted and saved them as 256×256 pixels format for still activity detection.
3. For moving activity detection at first foreground, human detection is done by a Gaussian mixture model.
4. For still activity detection at first foreground, human detection is done using the Histogram of Oriented Gradient (HOG) features and a trained Support Vector Machine (SVM) classifier. The object detects unoccluded people in an upright position.
5. After object detection features are calculated by using boundary boxing and centroid of the detected object of at most 100 images of a video.
6. After feature extraction, classification is done for moving activity from the predefined threshold which is set from the previous experience of the sample set.
7. For still activity after human object detection, those are saves as a gray image which is converted to a binaryimage for frame subtraction.
8. After frame subtraction features are extracted and predefined threshold which is set from the previous experience of sample set is usedin order to identify the activity.


Boundary boxing, centroid calculation
Input video stream with foreground objects
Gaussian Mixture Model
HOG Features and SVM Classifier
Segmentation and Resizing of image
Action Result
Classification
Feature extraction for upper body mass change calculation
Frame subtraction
Edge detection
Boundary boxing, centroid calculation and unwanted pixel removal

Complete flow diagram of the proposed approach

The entire task in this thesis, i.e. video reading, segmentation of the region of interest, feature extraction, data collection, classification is done by Matlab R2017a.

3.1 Segmentation and Resizing Images

The mostly static camera is used to capture the video both in KTH dataset and Weizmann dataset. There is 25 different person's video from KTH dataset, andnine different person's video from Weizmann dataset is recorded for each activity. In this project work, we have segmented each video according to its frames rate. If in any video number of frame is greater than 200, then only first 100 frames are stored in a file of size 256×256 pixels for further work.

Resized image

3.2 Foreground Detection

Here two different techniques are used for foreground detection. For moving object detection, Gaussian Mixture Model (GMM) is used. Since in Matlab Gaussian Mixture Model is unable to detect a still object, HOG feature extraction and SVM are combined for still object detection.

3.2.1 Gaussian Mixture Model (GMM)[20][10]

For foreground masking Gaussian Mixture Model is used and returned as a binary mask. In statistics, a mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population. However, while problems associated with "mixture distributions" relate to deriving the properties of the overall population from those of the sub-populations, "mixture models" are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population identity information.

A typical non-Bayesian Gaussian mixture model looks like this:

K = number of mixture components

N = number of observations

$\varphi_{i=1....K}$ = ***mixture weight, i.e., prior probability of a particular component i***

φ= ***K-dimensional vector composed of all the individual*** $\varphi_{1....K}$***; must sum to 1***

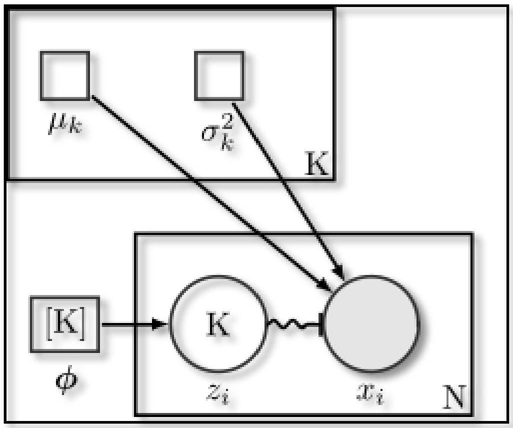
$\mu_{i=1.....k}$ ***=mean of component i***

$\sigma^2_{i=1....K}$ ***= variance 0f component i***

$\theta_{i=1....K} = \{\mu_{i=1.....k}, \sigma^2_{i=1....K}\}$

$z_{i=1.... N} \sim$ ***Categorical*** (φ)

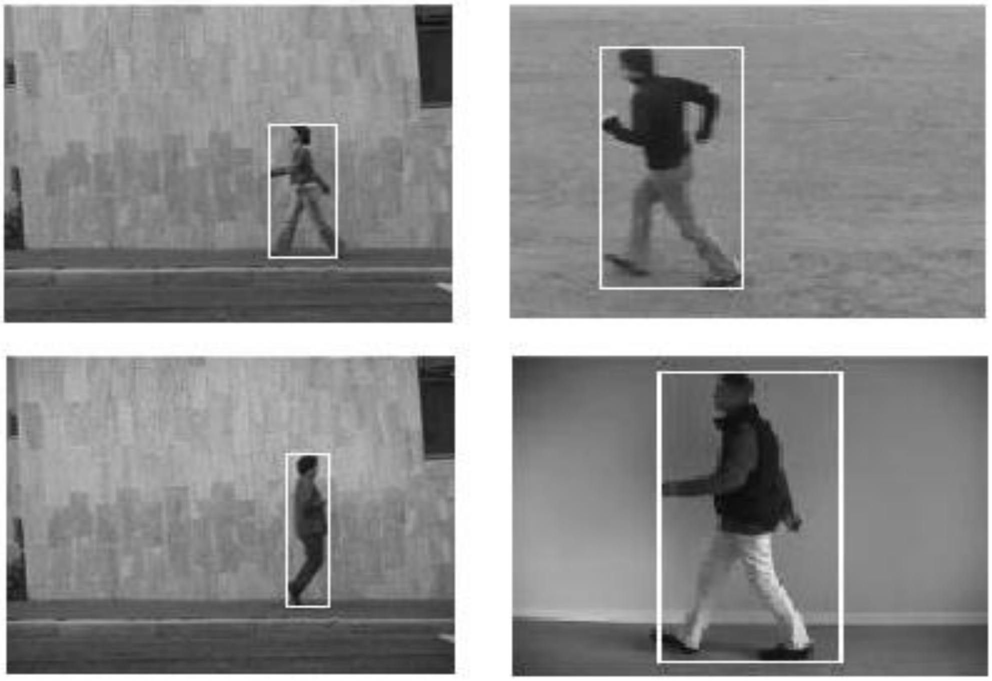
$x_{i=1....N} \sim \mathcal{N}(\mu_{zi}, \sigma^2_{zi})$



Non-Bayesian categorical mixture model using plate notation

Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size *K*.

In this project, I use 50 video frames for training background model with 3 Gaussian modes in the mixture model and the initial mixture model variance for “uint8” image data type is 30×30. The threshold to determine background model is 0.07.



Detected Human Beings

3.2.2 HOG features and SVM classifier [11]

HOG stands for Histograms of Oriented Gradients. HOG is a type of “feature descriptor.”The intent of a feature descriptor is to generalize the object in such a way that the same object (in this case a person) produces as close as possible to the same feature descriptor when viewed under different conditions. This makes the classification task easier.

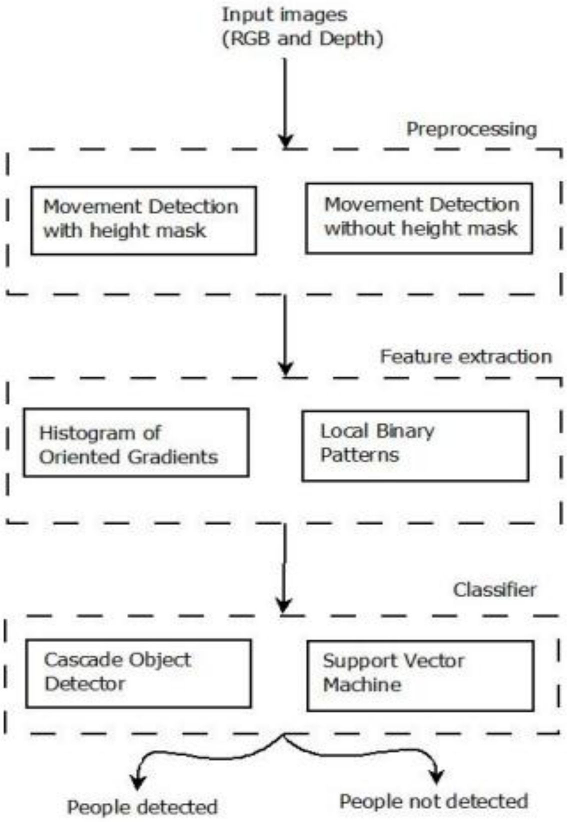
The creators of this approach trained a Support Vector Machine (a type of machine learning algorithm for classification), or “SVM,” to recognize HOG descriptors of people.

The HOG person detector is fairly simple to understand (compared to SIFT object recognition, for example). One of the main reasons for this is that it uses a “global” feature to describe a person rather than a collection of “local” features. This means that the entire person is represented by a single feature vector, as opposed to many feature vectors representing smaller parts of the person.

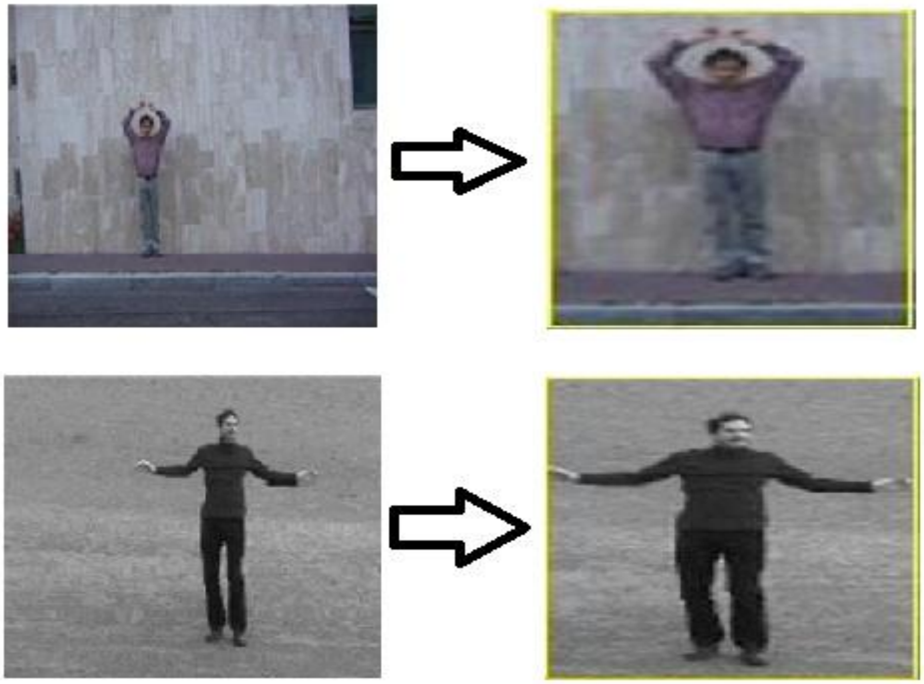
The HOG person detector uses a sliding detection window which is moved around the image. At each position of the detector window, a HOG descriptor is computed for the detection window. This descriptor is then shown to the trained SVM, which classifies it as either “person” or “not a person”.

To recognize persons at different scales, the image is sub-sampled to multiple sizes. Each of these sub-sampled images is searched.

In the project pre-defined “peopleDetector “method is used for people detection, which is created using HOG feature and SVM classifier. Human is detected from the frames and detected part is stored for feature extraction.



Flow diagram of working of peopleDetector [26]



Frames and Their Corresponding Detected Human

3.3 Feature Extraction

Two types of features are extracted for this project. One of them is for moving activity and one of them for still activity.

Terminologies

3.3.1 Manhattan Distance [1]:

The Manhattan distance function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

The formula for this distance between a point ***X*** = (x_1, x_2, etc.) and a point ***Y*** = (y_1, y_2, etc.) is:

$$D = \sum_{i=1}^{n} |x_i - y_i|$$

Where n is the number of variables, and x_i and y_i are the values of the ith variable, at points ***X*** and ***Y*** respectively.

3.3.2 Centroid

In mathematics, the centroid or geometric center of a plane figure is the arithmetic mean position of all the points in the shape. In image processing, the mean of all the coordinates of the labelled object both x and y coordinates, call this centroid. To calculate the centroid of an object, at first, a rectangular box is created around that box, which known as boundary boxing. Calculation of centroid from the image as follows.

- ***Label connected components in a binary image:*** it is used to compute a matrix of the same size as the input image. The matrix contains labels for the connected objects in the image. The number of connectedobjects is by default 8. The elements of the matrix are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object; the pixels labeled 2 make up the second object and so on.

Algorithm *[27][28]*

1. Run-length encodes the input image.
2. Scan the runs, assigning preliminary labels and recording label equivalences in a local equivalence table.
3. Resolve the equivalence classes.
4. Relabel the runs based on the resolved equivalence classes.

Example

This example illustrates using 4-connected objects. Notice objects 2 and 3; with 8-connected labeling, bwlabel would consider these a single object rather than two separate objects.

```
BW = [1     1     1     0     0     0     0     0
    1     1     1     0     1     1     0     0
1     1     1     0     1     1     0     0
     1     1     1     0     0     0     1     0
     1     1     1     0     0     0     1     0
     1     1     1     0     0     0     1     0
     1     1     1     0     0     1     1     0
     1     1     1     0     0     0     0     0];

L = bwlabel(BW,4)


L =
    1     1     1     0     0     0     0     0
    1     1     1     0     2     2     0     0
    1     1     1     0     2     2     0     0
    1     1     1     0     0     0     3     0
    1     1     1     0     0     0     3     0
    1     1     1     0     0     0     3     0
    1     1     1     0     0     3     3     0
    1     1     1     0     0     0     0     0

[r,c] = find (L==2);
```

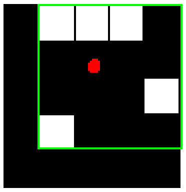
```
rc = [r c]

rc =
  2     5
  3     5
  2     6
   3     6
```

- **Measure properties of image regions[30]:**

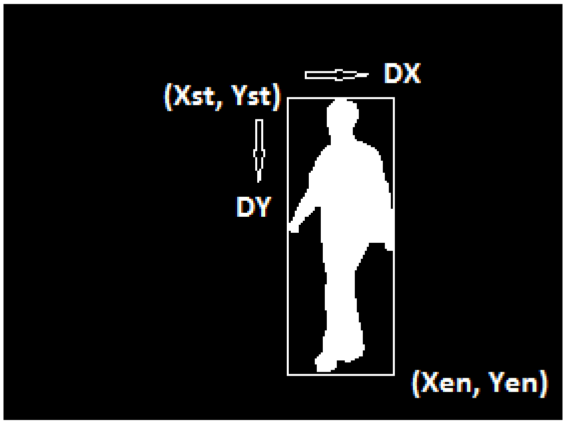
  It measures a set of properties for each labeled region in the label matrix calculated above for each 8-connected component (object) in the binary image. Positive integer elements of label matrix correspond to different regions. For example, the set of elements of the matrix equal to 1 corresponds to region 1; the set of elements of the equal to 2 corresponds to region 2; and so on. The return value is stored in a structure.

  Center of the mass of the region, is returned as a 1-by-*Q* vector. The first element of Centroid is the horizontal coordinate (or *x*-coordinate) of the center of mass. The second element is the vertical coordinate (or *y*-coordinate). All other elements of Centroid are in order of dimension. This figure illustrates the centroid and bounding box for a discontinuous region. The region consists of the white pixels; the green box is the bounding box, and the red dot is the centroid.



3.3.3 Boundary boxing (BBOX)[6]

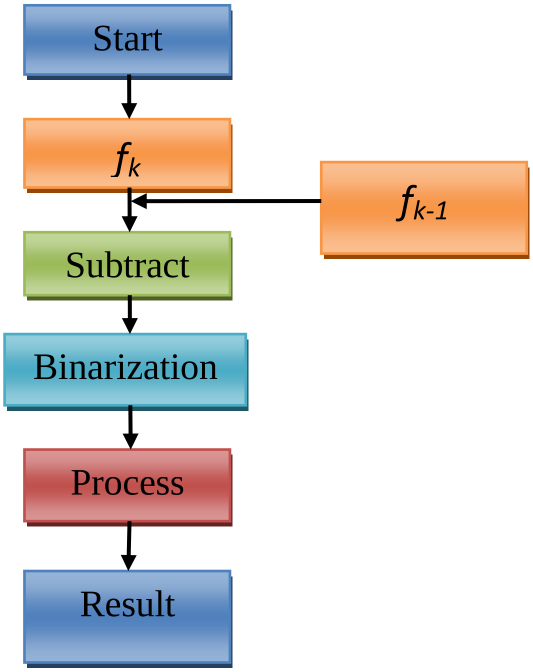
BBOX is the rectangle immediately surrounds the foreground silhouette, which is represented by (X_{ST}, Y_{ST}, *DX, and DY*). Where (X_{ST}, Y_{ST}), *DX*, and *DY* are the top-left point, width and height of the silhouette respectively.



BBOX

3.3.4 Frame Subtraction

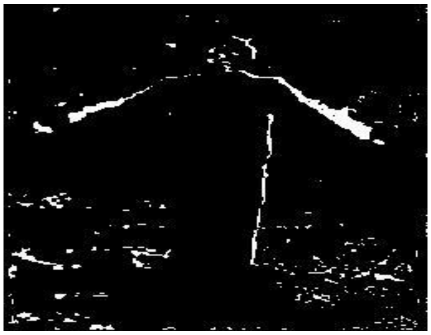
Frame difference calculates the difference between two frames at every pixel position and stores the absolute difference. It is used to visualize the moving objects in a sequence of frames. For still activities recognization, frame subtraction method is used. After detecting the human body from frames, we need to remove the still background. To do that first RGB images are converted to a binary image, and after that, any frame is subtracted from its previous frame to get the foreground moving part. And then centroid is calculated from that image.



Flowchart for frame difference method

Images, given below, are the examples of the present work.

Frame 1 Frame 2

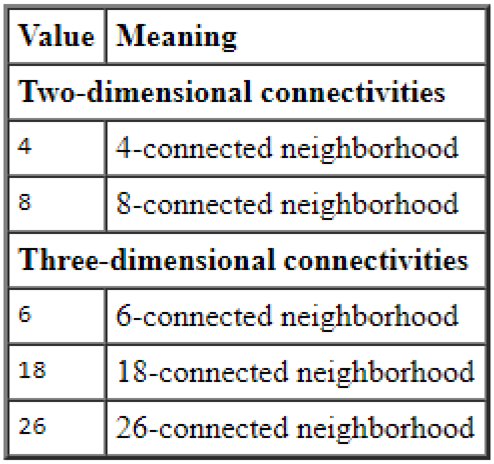


Frame 1 -- Frame 2

3.3.5Removal of unwanted small objects:

After frame subtraction, the subtracted binary image contains some unwanted small object in the background. These small objects create a problem in order to calculate the centroid of the foreground object. There are several approaches for removal of unwanted small objects. In this project, we have used the following procedure.

- ***Binary area open*** *[29]****:*** It removes all unwanted blobs. To do that it uses frame subtracted images, which is obtained previously. It removes from a binary image of all connected components (objects) that have less than a predefined number of pixels, producing another binary image. The default connectivity is 8 for two dimensions, 26 for three dimensions,



Value	Meaning
Two-dimensional connectivities	
4	4-connected neighborhood
8	8-connected neighborhood
Three-dimensional connectivities	
6	6-connected neighborhood
18	18-connected neighborhood
26	26-connected neighborhood

Algorithm:

The basic steps are:

1> Determine the connected components.

```
L = bwlabeln(BW, CONN);
```

2> Compute the area of each component.

```
S = regionprops(L, 'Area');
```

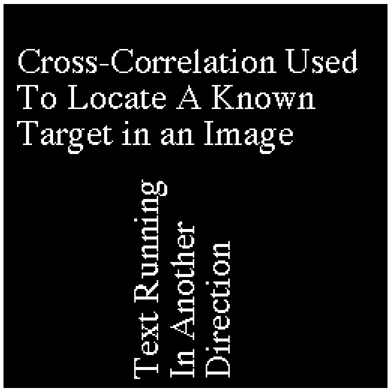
3> Remove small objects.

```
bw2 = ismember(L, find([S.Area] >= P));
```

Example: Remove all objects containing fewer than 40 pixels in an image.

- Read in the image and display it.

```
bw = imread('text.tif');
imshow(bw)
```

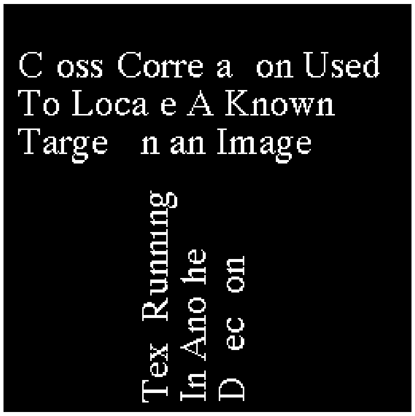


Cross-Correlation Used
To Locate A Known
Target in an Image

Text Running
In Another
Direction

1. Remove all objects smaller than 40 pixels. Note the missing letters.

```
bw2 = bwareaopen(bw,40);
figure, imshow(bw2)
```



3.3.6 Edge Detection [22]

Edge detection is the first step to recover information from images. Edges are the significant local changes of intensity in an image. Edges typically occur on the boundary between two different regions in an image. Edges also can be defined as discontinuities in image intensity from one pixel to another. A typical edge detector has following steps: (a) it suppresses noise as much as possible, without destroying the true edges; (b) it applies a filter to enhance the quality of the edges in the image, (c) it determines which edge pixels should be discarded as noise and which should be retained, (d) it determines the exact location of an edge. An optimal edge detector should satisfy the following criteria: (a) the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise) as well as that of false negatives (missing real edges), (b) the edges detected must be as close as possible to the true edges, (c) must return one point only for each true edge point; that is, it minimizes the number of local maxima around the true edge created by noise.

3.3.6.1 Sobel edge detector

The Sobel edge detector is one of the most commonly used image processing tool to detect edges from image. It has the following steps:

- **Gray Scale Conversion**
  In photography and computing, a grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as blackand- white, are composed exclusively of shades of gray, varying from black at the weakest intensity to

white at the strongest. To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value and 11% of the blue value.

- **Noise Reduction**

  The Sobel edge detector uses a filter based on the first derivative of a Gaussian because it is susceptible to noise exists in raw, unprocessed image data. Thus, at first, the raw image is convolved with a Gaussian filter. The result is a slightly blurred version of the original which is not affected by a single noisy pixel to any significant degree.

- **Gradient Computation**

  The edge may point to different directions.The first derivative is obtained,and the point of maxima is calculated. The Sobel algorithm uses an optimal edge detector based on a set of criteria which include finding the most edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and marking edges only a single edge exists for a minimal response.

- **Formulation[23]**

  The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical. If we define **A** as the source image, and $\mathbf{G}_x$ and $\mathbf{G}_y$ are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Where * here denotes the 2-dimensional signal processing convolution operation.

Since the Sobel kernels can be decomposed as the products of an averaging anda differentiation kernel, they compute the gradient with smoothing. For example, $\mathbf{G}_\mathbf{x}$ can be written as

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

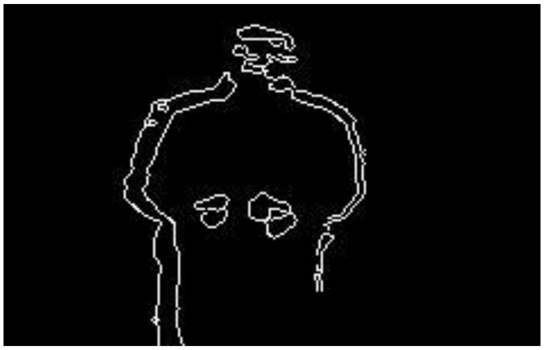
The *x*-coordinate is defined here as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

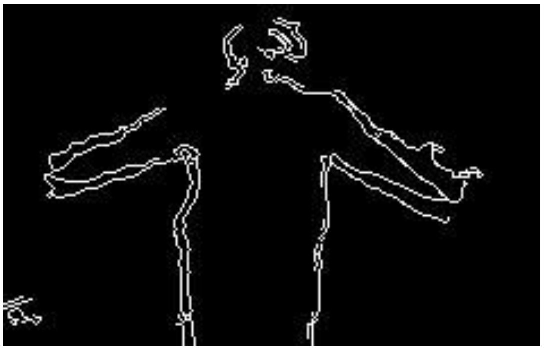
Using this information, we can also calculate the gradient's direction:

$$\mathbf{\Theta} = \operatorname{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Where, for example, **Θ** is 0 for a vertical edge which is lighter on the right side.



Hand Clapping



Hand Waving

3.3.7 Velocity Calculation

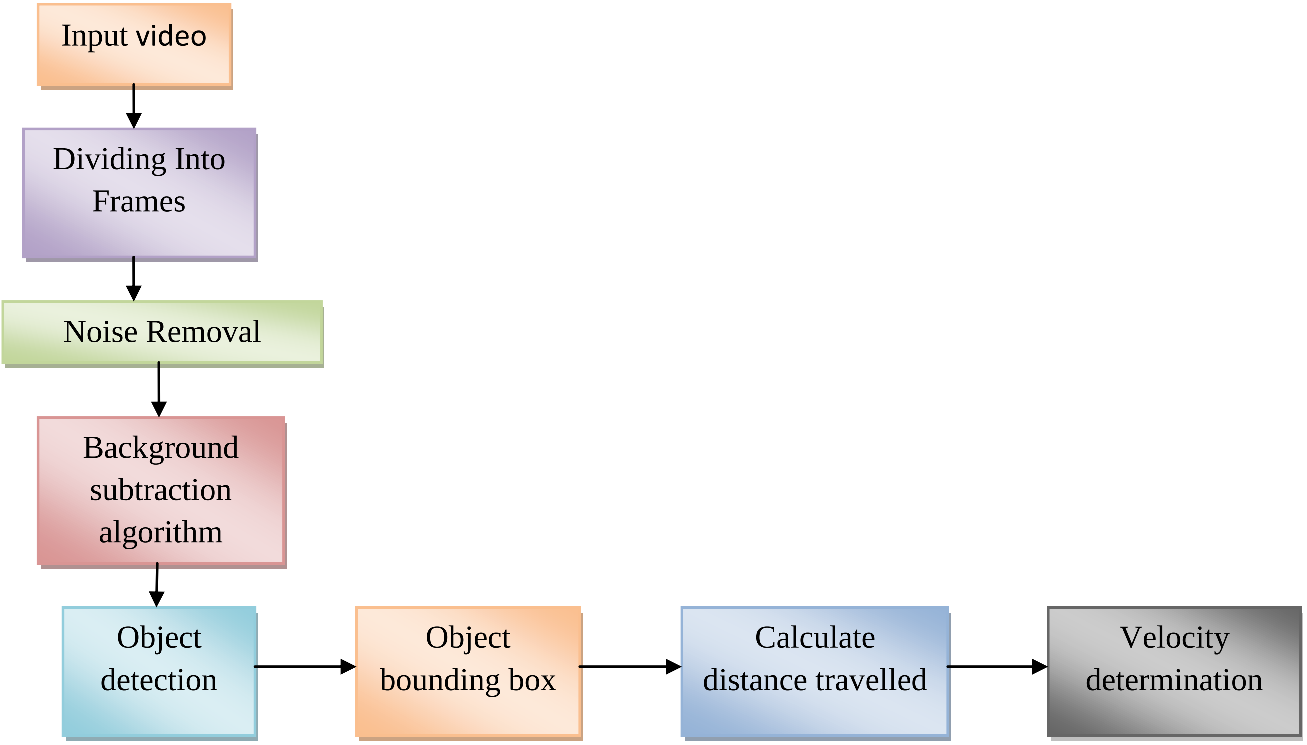
To separate “Running” and “Walking” activity velocity is calculated. Generally in running activity velocity of a person is higher than walking of that person. To calculate the velocity two things are calculated first,

1) Distance covered by a person from one side of the frame to another side.
2) In how many frames the above distance is covered.

By using the values of distance with respect to frame rate, the velocity of the object is defined. The defined velocity is of 2-dimension (since the camera is static).Velocity of moving object is determined using the distance traveled by the centroid to the frame rate of the video. The speed of moving object in the sequence frames is defined in pixels/second.

Velocity=distance/ number of frames.

The above Velocity value is usedin order to separate “Running” and “Walking” activity. The successive frames in a video of moving object are given in below figure.



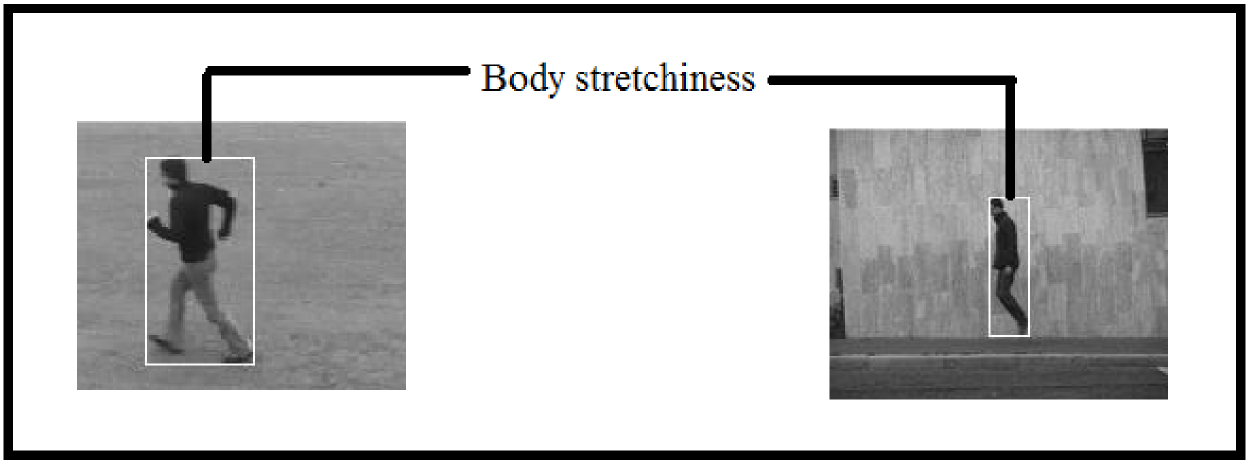
Flowchart of object determination

Object tracking, the primary application for security, surveillance and vision analysis. In this, a video is recorded using a digital camera. The recorded video frames are converted into individual frames. Noise is removedfrom the imported images using a median filter. The filtered images are used as input for the frame difference for the separation of foreground and background objects.

3.3.8 Body Stretchiness

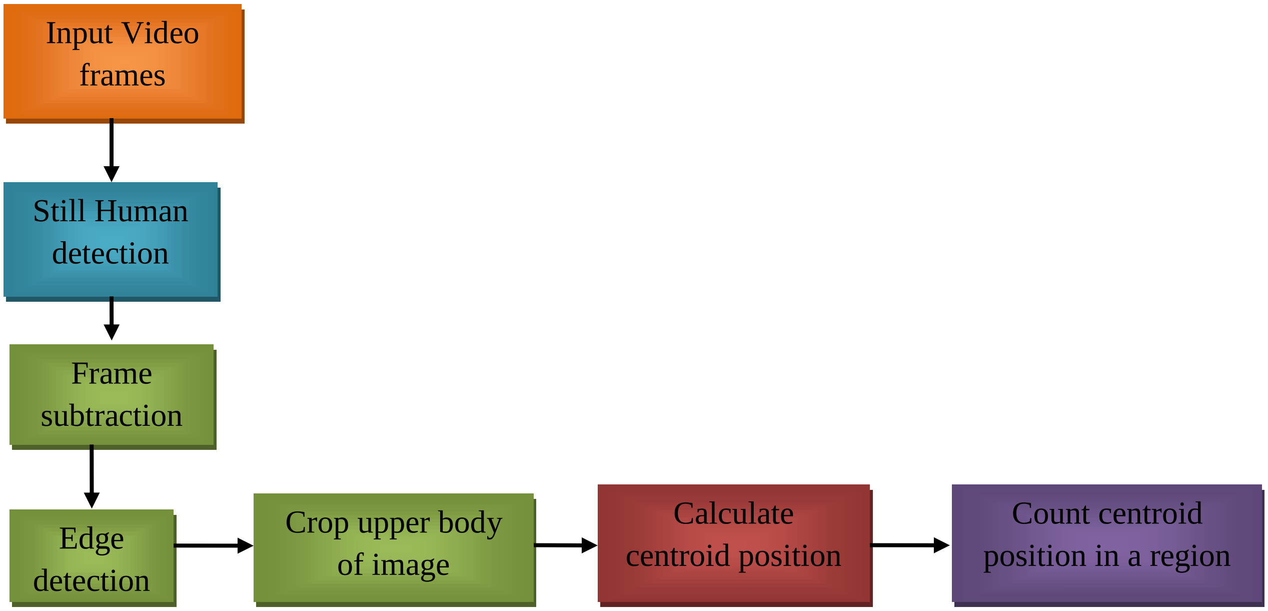
To differentiate another moving activity “Jumping” (from one place to another) from “Running” and “Walking” activities body stretchiness feature is calculated. This feature is calculated from previously saved boundary boxing of an object.

From the co-ordinate of the top left corner of the rectangular boundary and its width towards the horizontal axis, body stretchiness is calculated.



3.3.9 Upper body mass change

To separate two activities “Hand Waving” and “Hand Clapping” this feature is extracted. From the detection of the whole human body, only the upper portion of the body is identified and from that how many times the centroid is placed within that particular region is calculated and used for classification



Flow diagram of calculation of upper body mass change

CHAPTER 4

4 Human Activity Classifications

Here total five types of activity are classified. The events are running, walking, jumping, hand waving and hand clapping.

4.1 Separate Moving and Still Activities

Since Gaussian Mixture Model is unable to identify the still person soat first moving and still activities are separated by calculating a number of frames identified for human detection by Gaussian Mixture Model. If the number is less than ten then clearly the activity is still activity as in any video number of frames is more than 30.

4.2 Separate “Running” from “Walking”

Since the velocity for the running of a person is comparatively higher than walking of that person. So the velocity is calculated from sample video set, and a threshold is determined from that to classify “running” and “walking” activity.

	Name	Number of the frame to cross one side of the frame to another side(num)	Velocity=distance/num
Weizmann dataset	Moshe	20	7.5198
	Shahar	22	6.6918
KTH dataset	person1-run1	14	10.6225
	person1-run2	19	7.646
	person1-run3	12	11.8199
	person1-run4	14	10.3255
	person2-run1	18	8.2296
	person2-run2	33	4.0282
	person2-run3	26	5.7572
	person2-run4	32	4.6281

Table 1: Result of Sample data for “Running”

	Name	Number of frames to cross one side of the frame to another side(num)	Velocity=distance/num
Weizmann dataset	Moshe	37	4.05
	Shahar	40	3.8173
KTH dataset	person1-walk1	33	4.4574
	person1-walk2	48	3.17
	person1-walk3	33	4.5782
	person1-walk4	44	3.5211
	person2-walk1	28	5.3168
	person2-walk2	79	1.5959
	person2-walk3	36	4.1217
	person2-walk4	43	3.5118

Table 2: Result of Sample dataset for "Walking"

From the above result, threshold is calculated and fixed for value Threshold (velocity) = 5.5

4.3 Separate "Jumping" from "Running" and "Walking"

To separate jumping from running and walking body-stretchiness feature is used because the body stretchiness for running and walking f a person is much higher than jumping of that person. A threshold is calculated from the sample set, which is used for recognizing jumping activity.

		Body stretchiness		
	Name	Running	Walking	Jumping
Weizmann dataset	Moshe	37	39	28
	Shahar	40	39	22
KTH dataset	person1-action1	70	55	Do not have sample data set
	person1-action2	49	42	
	person1-action3	61	53	
	person1-action4	82	77	
	person2-action1	63	56	
	person2-action2	74	67	
	person2-action3	51	52	

	person2-action4	82	89	

Table 3: Maximum body stretchiness for “Running”

From the above sample data set a threshold for jumping activity is set and

Threshold =30.

4.4 Separate “Hand Waving “and “Hand Clapping”

Depending on the centroid position on upper body mass these two still activities are recognized. Generally in “Hand Waving” number of time occurred of the centroid in the upper portion of human body is more than “Hand Clapping.”

	Name	Hand Waving	Hand Clapping
KTH Dataset	person1-action1	33.3333	15
	person1-action2	36.4865	9.901
	person1-action3	39.5349	4.8387
	person1-action4	9.1837	14.8515
	person2-action1	47.9452	17.8218
	person2-action2	37.6238	8.9109
	person2-action3	47.5248	5.9406
	person2-action4	47.3684	10.8911

Table 4: still action on KTH database

From the above sample set threshold is calculated. And the threshold is

Threshold= 30 to 68 for hand waving

Otherwise, hand clapping

Thus all five human activities are classified.

CHAPTER 5

RESULT

[Both Weizmann and KTH datasets are used for this project. There are 100 to 110 videos for each activity. Out of 110 videos 8 to 10 videos are used for training and remaining videos are used for testing.]

The class of actions includes running, walking, jumping, hand waving and hand clapping according to our consideration. Thus to the evaluation of the proposed action recognition, we use two publicly available datasets.

5.1 Datasets

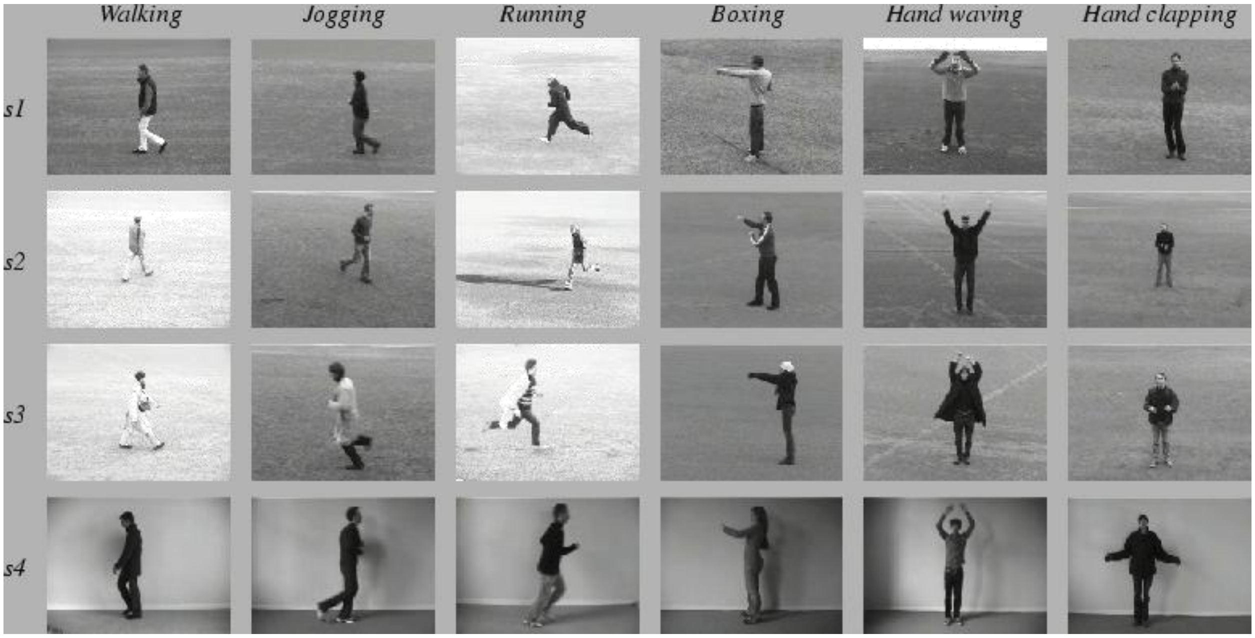
Weizmann [14] and KTH [15] datasets are used for this project.

5.1.1 Weizmann Dataset:

This is a very common dataset; many state of the art approaches use the dataset for their purpose and evaluate performance on it, which allows easy comparison. The database contains 90 low-resolution (180 x 144, deinterlaced 50 fps) video sequences showing nine different people, each performing 10 natural actions such as "run," "walk," "skip," "jumping-jack" (or shortly "jack"), "jump-forward-on-two-legs" (or "jump"), "jump-in-place-on-two-legs" (or "pjump"), "gallop sideways" (or "side"), "wave-two-hands" (or "wave2"), "wave one- hand" (or "wave1"), or "bend."[14]

5.1.2 KTH Dataset:

This database is also widely used for Human Activity Recognition. The current video database containing six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors *s1*, outdoors with scale variation *s2*, outdoors with different clothes *s3* and indoors *s4* as illustrated below. Currently the database contains 2391 sequences. All sequences were taken over homogeneous backgrounds with a static camera with *25*fps frame rate. The sequences were downsampled to the spatial resolution of *160x120* pixels and have a length of four seconds in average. All sequences are stored using AVI file format. [15]



http://www.nada.kth.se/cvap/actions/actions.gif

5.2 Machine Configuration

System- HP Notebook 15ac024TX

Processor: Intel® Core™ i3-4005U CPU @ 1.70GHz
RAM: 4.00 GB System type:
Windows 8.1 Pro© 2013, 64-bit Operating System, x64-based processor

Tools – MATLABR2017a

5.3 Evaluation strategy

For evaluation of the performance of the proposed methodology confusion matrix and Misclassification rate (MCR) are used.

5.3.1 Confusion Matrix:

Confusion matrix gives a clear knowledge about the actual and wrong classification of any classifier. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

It is a particular kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table) [16]. For example in Table 6 out of 10 ‘hand waving’ videos of Weizmann dataset our method recognize 8 videos ad ‘hand waving’ and other 2 videos as ‘hand clapping’. In the case of Weizmann dataset, we have confusion matrix only for four activities ‘running’, ‘walking’, ‘jumping’, ‘hand waving’. Here ‘hand waving’ is confused with ‘hand clapping’. the other hand, for KTH dataset, we have confusion matrix for ‘running’, ‘walking’, ‘hand waving’, ‘hand clapping’. ‘Hand clapping’ and ‘Hand waving’ are confused with each other and ‘running’ is confused with ‘walking’.

The result of the classification of human activities is shown by confusion matrix.

ACTION	RUN	WALK	HAND WAVING	HAND CLAPPING
RUN	0.92	0.03	0.0	0.0
WALK	0.08	0.97	0.0	0.0
HAND WAVING	0.0	0.0	0.78	0.21
HAND CLAPPING	0.0	0.0	0.22	0.79

Table 5: confusion Matrix on KTH Database

ACTION	RUN	WALK	JUMP	HAND WAVING
RUN	1.0	0.0	0.0	0.0
WALK	0.0	1.0	0.0	0.0
JUMP	0.0	0.0	1.0	0.0
HAND WAVING	0.0	0.0	0.0	0.8

Table 6: Confusion Matrix on Weizmann Dataset

5.3.2 Misclassification Rate (MCR)

Definition:

Let X be a feature space with a finite number of elements. Moreover, let C be a set of classes, let y: X → C be a classifier, and let c be the target concept to be learned. Then the true misclassification rate, denoted as Err ∗ (y), is defined as follows:[17]

$\mathrm{Err} * (y) = |\{x \in X: y(x) \neq c(x)\}| / |X|$

Our proposed approach is entirely dependent on the correctness of segmentation and foreground detection. If Gaussian mixture model is unable to detect moving human then misclassification rate increases. On the other hand if people detector is unable to detect the boundary boxing around human body then misclassification rate will increase for activity 'hand waving' and 'hand clapping'. In Weizmann dataset misclassification rate between 'hand waving' and 'hand clapping' is 5.56% and in KTH dataset misclassification rate is 13.5%.

5.4 Analysis

Confusion matrix assured the effective accuracy rate of our proposed technique. The misclassification has taken place in case of both 'running', 'walking' and 'hand waving', 'hand clapping' for KTH dataset. And in Weizmann dataset it only occurs for 'hand waving' and 'hand clapping'. Overall performance is 94.44% for

Weizmann dataset and 86.5% for KTH dataset which will be observed from Table 5 and Table 6.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter, we look back at the problem defined in the introduction and summarize the work done in this project towards solving it. Conclusions are drawn from the work, particularly the results presented in the last chapter. Finally, possible future work is suggested.

6.1 Conclusion

This thesis has shown how a combination feature extraction, evaluation, and selection can work together to provide a high-quality data set for use with an inference engine.

Feature extraction is not necessarily a difficult problem. In fact, as the feature evaluation step has shown, many of the best features turn out to be model data or elementary functions of it. For some activities, more complex features are needed.

Feature selection can be made using a variety of methods. A few such were attempted in this project and the idea behind choice turns out to be sound.

To summarize, we may draw the following conclusions:

- Feature extraction is effective and relatively simple.
- Feature evaluation can be done automatically by statistical means.
- Feature subset selection is sensible and useful.
- Extraction, evaluation, and selection work well together.

Especially notable is how well the combination of feature extraction and selection works. With careful selection, even simple features may contribute significantly to recognition results [9].

6.2 Future Work

The feature extraction step can be expanded almost without limits. It is indeed possible to apply more advanced statistical means to extract more complex features. However, it should be kept in mind that the evaluation tends to suggest that complex features are not generally excellent.

The evaluation and selection step could also benefit from more advanced statistics and information analysis. Sadly, it has been shown that the problem is computationally hard, and all efficient algorithms are bound to make use of heuristics or estimations. A relatively simple method to improve on the algorithms used would be to use a more dynamic subset size so that activities that are poorly recognized are allowed to use more features.

The most significant single point of improvement upon the recognition results could well be using an inference engine with intrinsic time modeling, for instance, a state-space model such as a Hidden Markov Model [9].

REFERENCES

[1] A. Bulling, U. Blanke and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors", ACM Computing Surveys (CSUR), vol. 46, no. 3, pp. 1-33, 2014.

[2] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Commun. Surveys Tuts., vol. 15, no. 3, pp. 1192–1209, 2013.

[3] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in Proc. 23rd Int. Architecture of Computing Systems Conf., Hannover, Germany, 1-10, 2010.

[4] X. Su, H. Tong, and P. Ji, “Activity recognition with smartphone sensors,” Tsinghua Science and Technology, vol. 19, pp. 235–249, June 2014.

[5] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery,” IEEE Pervasive Computing, vol. 9, no. 1, pp. 48–53, 2010.

[6]Satyabrata Maity, Dr. Amlan Chakrabarti, and Dr. Debotosh Bhattacharjee ,”*Robust Human Action Recognition using AREI Features and Trajectory Analysis from Silhouette Image Sequence”*

[7] Sermetcan Baysal, Mehmet Can Kurt and Pınar DuyguluBilkent University, Department of Computer Engineering, 06800, Ankara, Turkey*,”Recognizing Human Actions Using Key Poses”*

[8]Diogo Carbonera Luvizon _ Hedi Tabia David PicardETIS {UMR CNRS 8051 {ENSEA { Universit_e Paris Seine / Universit_e de Cergy-Pontoise fdiogo.luvizon, hedi.tabia, picardg@ensea.fr*,”Learning features combination for humanaction recognition from skeleton sequences*”

[9] Sebastian Brannstorm,”*Extraction, Evaluation and Selectionof Motion Features for Human Activity Recognition Purposes*”, TRITA-CSC-E 2006:028, ISRN, KTH/CSC/E--06/028—SE, ISSN-1653-5715

[10]https://in.mathworks.com/help/vision/ref/vision.foregrounddetector-system-object.html(Accessed on 20.3.2018)

[11]https://in.mathworks.com/help/vision/ref/vision.peopledetector-system-object.html (Accessed on 23.3.2018)

[12]https://in.mathworks.com/matlabcentral/answers/28996-centroid-of-an-image(Accessed on 24.3.2018)

[13]https://in.mathworks.com/discovery/edge-detection.html(Accessed on 27.3.2018)

[14] http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html (Accessed on 24.3.2018)

[15] http://www.nada.kth.se/cvap/actions/ (Accessed on 24.3.2018)

[16] https://en.wikipedia.org/wiki/Confusion_matrix (Accessed on 24.5.2018)

[17] http://www.uni-weimar.de/medien/webis/teaching/lecturenotes/machine-learning/unit-en-performance-measures.pdf (Accessed on 21.5.2018)

[18] http://www.rroij.com/open-access/human-activity-recognition-challenges-and-process-stages-.pdf (Accessed on 19.5.2018)

[19] https://en.wikipedia.org/wiki/Pattern_recognition (Accessed on 24.5.2018)

[20] https://en.wikipedia.org/wiki/Mixture_model (Accessed on 20.2.2018)

[21] https://www.slideshare.net/srikanthgadam/human-activity-recognition (Accessed on 24.5.2018)

[22] https://www.ijert.org/download/2131/moving-object-detection-and-velocity-estimation-using-matlab (Accessed on 24.05.2018)

[23] https://en.wikipedia.org/wiki/Sobel_operator (accessed on 24.05.2018)

[24]https://www.engineersgarage.com/articles/image-processing-tutorial-applications(Accessed on 19.5.2018)

[25]https://www.google.co.in/search?q=pattern+recognition&rlz=1C1CHBD_enIN770IN770&source=lnms&tbm=isch&sa=X&ved=_AUICygC&biw=1366&bih=662#imgrc=0ahUKEwiol9rJk57bAhXBMI8KHXpcDgMQ_pY3tfVuc12fFCM:
(Accessed on 19.5.2018)

[26] https://www.ehu.eus/documents/3444171/4484752/61.pdf (Accessed on 10.5.2018)

[27] Haralick, Robert M., and Linda G. Shapiro. *Computer and Robot Vision, Volume I.* Addison-Wesley, 1992. pp. 28-48

[28] https://edoras.sdsu.edu/doc/matlab/toolbox/images/bwlabel.html (Accessed on 25.5.2018)

[29] https://edoras.sdsu.edu/doc/matlab/toolbox/images/bwareaopen.html (Accessed on 25.5.2018)

[30] https://edoras.sdsu.edu/doc/matlab/toolbox/images/regionprops.html#254834 (Accessed on 25.4.2018)

[31] https://dl.icdst.org/pdfs/files/da090a75f2b3c3179de82d428b33ef4d.pdf (Accessed on 27.5.2018)