

# HUMAN ACTIVITY DETECTION FROM VIDEOS

*by*

**R. SUCHARITHA** 2013103026  
**ANIRUDH RAVICHANDRAN** 2013103055

*A project report submitted to the*

**FACULTY OF INFORMATION AND  
COMMUNICATION ENGINEERING**

*in partial fulfillment of the requirements for  
the award of the degree of*

# BACHELOR OF ENGINEERING

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE AND  
ENGINEERING**

**ANNA UNIVERSITY, CHENNAI – 25**

MAY 2017

## **BONAFIDE CERTIFICATE**

Certified that this project report titled **HUMAN ACTIVITY DETECTION FROM VIDEOS** is the *bonafide* work of **R. SUCHARITHA (2013103026)** and **ANIRUDH RAVICHANDRAN (2013103055)** who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.

**Place:** Chennai

**Dr. S. CHITRAKALA**

**Date:**

Associate Professor

Department of Computer Science and Engineering

Anna University, Chennai – 25

**COUNTERSIGNED**

Head of the Department,  
Department of Computer Science and Engineering,  
Anna University Chennai,  
Chennai – 600025

## **ACKNOWLEDGEMENT**

We would like to extend our immense gratitude to our project guide, Prof Dr. S. Chitrakala for her perpetual support and able guidance which was instrumental in taking the project to its successful conclusion. We would like to thank our project committee members Prof Dr. Valli, Prof Dr. S. Sudha and Prof Dr. B. Velammal for constantly motivating us and identifying the areas of improvement on the project. Finally, we would like to thank the Head of the department, Prof Dr. D. Manjula for providing a conducive environment and amenities to facilitate our project work.

**R. SUCHARITHA**

**ANIRUDH RAVICHANDRAN**

## **ABSTRACT**

Human actions are very complex and requires a continuous movement over a period. Understanding human activities to help people in a variety of applications is the main goal of this project. We use the KTH dataset to train our system which Recognizes the correct activity using a KNN classifier. The KTH dataset has the following actions:

Jumping, Boxing, Walking, Skipping and Jogging.

The aim is to process the video and to correctly identify these activities mentioned above. One of the major issues faced in our system is the segmentation and recognition of a sequence of human activities from continuous unsegmented visual data.

We train the system by feeding in video footage of different actions. A database is created and the KNN classifier is used to generate class labels. The human actions are analysed by a pre-processing mechanism that converts the video into frames and then extracts the features from it. The features extracted are Histogram Oriented Gradients(HOG). The features are measures of edge gradients of the person in the video footage. When a test footage is fed into the system, a pop up recognizes the correct action by identifying the correct class labels.

The performance is measured by various metrics such as accuracy, f-score, recall and precision.

The applications of this system reach wide areas in surveillance, security, and forensics.

# **ABSTRACT**

# TABLE OF CONTENTS

<b>ABSTRACT – ENGLISH</b>	iii
<b>ABSTRACT – TAMIL</b>	iv
<b>LIST OF FIGURES</b>	viii
<b>LIST OF TABLES</b>	x
<b>LIST OF ABBREVIATIONS</b>	xi
<b>1 INTRODUCTION</b>	1
1.1 General Overview	1
1.2 About The Project	2
1.2.1 Objectives	3
1.2.2 Problem Statement	3
1.2.3 Need for Human Activity Recognition	4
1.2.4 Challenges in Video Processing	4
1.2.5 Overview of The Thesis	5
<b>2 RELATED WORK</b>	6
2.1 Human Activity Modeling	6
2.2 Temporal Activity Segmentation	7
2.3 HOG Feature Extraction	8
2.4 Observation from Literature Survey	10
<b>3 REQUIREMENTS ANALYSIS</b>	12
3.1 Functional Requirements	12
3.2 Non-Functional Requirements	12
3.2.1 User Interface	12

3.2.2	Software . . . . .	13
3.2.3	Performance . . . . .	13
3.3	Constraints and Assumptions . . . . .	13
<b>4</b>	<b>SYSTEM DESIGN . . . . .</b>	<b>14</b>
4.1	Proposed System . . . . .	14
4.2	System Architecture . . . . .	14
4.3	List of Modules . . . . .	15
4.4	System Modules . . . . .	16
4.4.1	UseCase Diagram . . . . .	16
4.4.2	Sequential diagram . . . . .	16
4.5	Modules . . . . .	17
4.5.1	Preprocessing . . . . .	17
4.5.2	Gradient Calculation . . . . .	19
4.5.3	Block and Cell formation . . . . .	20
4.5.4	Histogram Formation . . . . .	20
4.5.5	Normalisation . . . . .	21
4.5.6	KNN Classifier . . . . .	21
<b>5</b>	<b>SYSTEM DEVELOPMENT . . . . .</b>	<b>23</b>
5.1	System Flow . . . . .	23
5.2	Algorithms . . . . .	23
5.2.1	Preprocessing . . . . .	23
5.2.2	HOG Feature Extraction . . . . .	24
5.2.3	Activity Detection . . . . .	28
<b>6</b>	<b>RESULTS AND DISCUSSION . . . . .</b>	<b>30</b>
6.1	Dataset . . . . .	30

6.2	Implementation details and discussion . . . . .	30
6.2.1	Evaluation . . . . .	32
6.2.2	Test Cases . . . . .	38
6.2.3	Performance Metrics . . . . .	43
<b>7</b>	<b>CONCLUSION . . . . .</b>	<b>44</b>
7.1	Conclusion . . . . .	44
7.2	Future Work . . . . .	44
	<b>REFERENCES . . . . .</b>	<b>46</b>



## LIST OF FIGURES

4.1	System Architecture . . . . .	15
4.2	UseCase Diagram . . . . .	17
4.3	Sequence Diagram . . . . .	18
4.4	Framing . . . . .	18
4.5	Pre processing . . . . .	19
5.1	Framming of videos . . . . .	24
5.2	Gradient Calculation in Y direction . . . . .	25
5.3	Gradient Calculation in X direction . . . . .	25
5.4	Iteration of blocks. . . . .	26
5.5	Iteration of cells. . . . .	27
5.6	Histogram Formation. . . . .	28
5.7	Final Result. . . . .	29
6.1	Recognised Boxing. . . . .	31
6.2	Recognised hand clapping. . . . .	32
6.3	Recognised hand waving. . . . .	33
6.4	Recognised jogging. . . . .	34
6.5	Recognised walking. . . . .	35
6.6	Accuracy Graph. . . . .	35
6.7	Precision, Recall and Fscore Graph. . . . .	36
6.8	Training Graph. . . . .	37
6.9	Testing Graph. . . . .	37
6.10	Test case for Boxing . . . . .	39
6.11	Test case for Hand Clapping. . . . .	40
6.12	Test case for Hand Waving. . . . .	41

6.13 Test case for Walking. . . . .	42
-------------------------------------	----

## **LIST OF TABLES**

6.1	Performance Matrix. . . . .	36
6.2	Training and Testing time. . . . .	36
6.3	Time Duration table. . . . .	38

## **LIST OF ABBREVIATIONS**

**GT** Greater than

**LT** Less than

**EQ** Equal to

**HOG** Histogram of oriented gradients

**HAR** Human activity recognition

**KNN** k-nearest neighbours

**TP** True Positive

**TN** True Negative

**FP** False Positive

**FN** False Negative

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 GENERAL OVERVIEW**

Human Activity Recognition(HAR) aims to recognize or understand human activity based on features extracted from human posture sequences in a video and identify the activity,here the focus is on interaction,activity inferred from identifying the activities of two persons.There are a lot of applications of HAR which includes human-machine interaction, intelligent video surveillance, sports event analysis, and content based video retrieval etc.Among these applications sports event analysis is our focus, where the interactions of the players needs to be identified.

Video processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams. Video processing techniques are used in television sets, VCRs, DVDs, video codecs, video players, video scalers and other devices. For examplecommonly only design and video processing is different in TV sets of different manufactures.Video processors are often combined with video scalers to create a video processor that improves the apparent definition of video signals. They perform the following tasks: deinterlacing aspect ratio control, digital zoom and pan, brightness/contrast/hue/saturation/sharpness, frame rate conversion and inverse-telecine, color point conversion, color space conversion, mosquito noise reduction, block noise reduction, etc.

## 1.2 ABOUT THE PROJECT

The main objective of the project is to identify the correct activity that is being performed which is jogging, boxing, clapping, hand waving and walking. In this project we are dealing with the KTH dataset that consist of vidoes of a single person doing these activities mentioned above. Intially each and every video is put through the preprocessing process which consists of framing, blurring and greying. The foreground is extracted from each and every frame. From every frame the RGB values are extracted. Then each frame undergoes the process of greying. In order to extract Histogram of Oriented Gradients (HOG), it is required to convert the RGB to a grey scale.

The Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. After conversion of the RGB image into grey scale, the gradient value is calculated in the X and Y direction. First the image is divided into a 16x16 blocks. Each of these blocks are further divided into 8x8 cells. Each and every pixel value from each cells are calculated. The histogram values are then calculated from these pixel values that have been obtained in the previous step. This process is repeated in every cells in every block of the image. After the feature vectors are obtained, the feature vectors undergoes the process of Normalisation. One of the major reason for Normalisation is because to removed the unwanted.

A feature repository is created using all the normalised feature

vector. Using this database, class labelling is done which is used in order to train a model that could be used for comparison with a testing video to identify the activity which is the goal of the project. Using the concept of KNN Classifier, we would be able to identify the correct activity that is being performed by the human in each video. The KNN Classifier uses the k-nearest neighbours algorithm. The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. The output depends on whether k-NN is used for classification or regression: In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors. In the testing phase, once again the HOG features are extracted and processed through KNN. It will try to find the most accurate and closest label from the already trained model in our system and eventually predicting the activity that is being performed.

### **1.2.1 Objectives**

The objective is to develop a system that identifies human activity that is being performed in a video.

1. Extraction of HOG features from a video
2. Using KNN classifier to identify the nearest possible feature.
3. Recognizing the activity being performed by the person in the video.

### **1.2.2 Problem Statement**

The system focuses on identifying the activity being performed in a continuous unsegmented video using the concept of human action recognition. Using the KTH dataset, HOG features are extracted after pre-processing of a video. Using the vector features, a model is trained

and labeled with five different activity namely jogging, clapping, skipping, running and boxing. During the testing phase, KNN classifier is used to find the closest possible label and the activity is identified.

Some of the issues faced were the compilation and execution time which has been overcome by using the KNN algorithm. The segmentation of continuous video footage is also another major issue.

### **1.2.3 Need for Human Activity Recognition**

Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agent. This research field has captured the attention of several computer science communities due to its strength in providing personalized support for many different applications such as medicine, human-computer interaction, or sociology. Traditionally, the task of observing and analyzing human activities was carried out by human operators, for example, in security and surveillance processes or the processes of monitoring a patients health condition. With the increasing number of camera views and technical monitoring devices, however, this task becomes not only more challenging for the operators but also increasingly cost-intensive, in particular, since it requests around-the-clock operation. In practice, for the case of home care, personnel deployment for such tasks often cannot be financially feasible.

### **1.2.4 Challenges in Video Processing**

Some of the challenges faced by a video processing system are as follows.

1. Segmenting and recognizing a sequence of human activities from



continuous unsegmented visual data.

2. Performing multiple task at the same time will make the activity recognition process much slower and difficult.

### **1.2.5 Overview of The Thesis**

Chapter 2 provides related works of Human activity/action recognition and HOG feature extraction.

Chapter 3 deals with the requirements and the requirements analysis for the proposed system.

Chapter 4 deals with the overall system architecture and the corresponding module descriptions.

Chapter 5 deals with system development for the system.

Chapter 6 shows the implementation and result of the proposed system.

Chapter 7 deals with conclusion and future enhancement.

## CHAPTER 2

### RELATED WORK

#### 2.1 HUMAN ACTIVITY MODELING

Many studies in human activity understanding have focused on recognizing repetitive or punctual activities from short manually partitioned visual data, which can be acquired from color or color-depth cameras. Instead of discussing supervised learning methods used to classify human activities at the model level, we focus on encoding spatio-temporal information at the feature level to distinguish temporal activity patterns.

This method, as in[6], encodes a human as a single point, which represents human locations in spatial dimensions. Another widely used human activity representation is based on articulated human body models, such as the skeleton model [7][8]. The third category of space-time representations employ a sequence of human shapes[9], including human contours and silhouettes [10], to model temporal activity patterns. Different from global human representations, LST features have attracted attention, due to the robustness to partial occlusion, slight illumination variation, and image rotation, scaling, and translation[8]. Because LST features are directly computed from raw visual data, they can avoid potential failures of preprocessing steps such as human localization and tracking. However, we address the task of continuous activity segmentation and recognition in unsegmented sequences.

A direct application of LST features to format time series generally

makes the segmentation problem intractable, because the raw LST features can contain a large number of elements in high-dimensional space. We bridge the divide between the continuous activity segmentation problem and the local human representation using LST features by introducing a new layer (i.e., block-level activity summarization) that projects the high-dimensional feature space to the low-dimensional activity space.

## 2.2 TEMPORAL ACTIVITY SEGMENTATION

Automatic segmentation of complex continuous activities is important, as intelligent robots deployed in human social environments receive continuous visual data from their on board perception systems. Without the capability of segmenting the continuous visual data into a temporal sequence of individual activities, it is impossible for robots to understand human behaviors and effectively interact with people. Previous continuous activity segmentation approaches can be generally grouped into three categories:

1. Heuristics
2. Optimization
3. Change point detection.

The first uses simple heuristics to segment human activities from continuous visual data. Another framework uses optimization, typically based on discriminative learning, to segment continuous human activities. The third category is based on change point detection. The earliest and best-known method is the cumulative sum control chart detector [10], which encodes a time series as piecewise segments of Gaussian means with noise. Given the satisfactory performance of the methods based on optimization or change point detection, they typically assume

xed boundaries of each activity event and, thus, are incapable of modeling gradual transitions between continuous activities in real-world situations. Different from previous continuous human activity segmentation methods that assume xed event boundaries, our objective is to explicitly model gradual transitions between temporally adjacent activities. We propose to apply temporal clustering to achieve this objective, which encodes each activity event as a fuzzy set with non xed boundaries, instead of segmenting visual data into disjoint events. In addition, the time series used in our algorithm is formulated by concatenating block-level human activity distributions.

There are two research problems different from temporal fuzzy segmentation. The rst problem is fuzzy recognition, which employs fuzzy methods to recognize activity states, i.e., to assign an activity category to a data instance. The second problem is background-foreground segmentation, which aims at localizing humans in the scene and spatially segmenting people from the background. The research partitions continuous data into events along the time dimension using temporal fuzzy clustering. The probabilistic method also derives fuzzy scores of events in the time dimension; such incrementally changing scores make temporal segmentation and activity recognition results accurate and stable.

### **2.3 HOG FEATURE EXTRACTION**

Real-time object detection has been a key technology in various application domains such as surveillance, automotive systems, and robotics. An important algorithm used in object detection systems, Histogram of Oriented Gradients (HOG) [1], has robustness to change of illumination and attains high computational accuracy in detection of

variously textured objects. Recent high-performance general-purpose processors can achieve real-time object detection at a heavy computational cost. However, the processor requires high power consumption and is therefore unsuitable for mobile systems under limited battery conditions. Consequently, a low-power and high-performance HOG feature extraction processor is necessary to widen the range of applications. Some FPGA implementations [3], [5], [6], [8], [9] and an FPGA-GPU architecture [4] have been proposed for real-time applications. HOG features are adaptable to widely various applications. Consequently, next-generation HOG feature extraction processors must provide higher expandability and higher performance. Therefore, our goal is to develop design techniques for a real-time HOG feature extraction processor for HDTV resolution video.

Real-time object detection has been a key technology in various application domains such as surveillance, automotive systems, and robotics. An important algorithm used in object detection systems, Histogram of Oriented Gradients (HOG) [1], has robustness to change of illumination and attains high computational accuracy in detection of variously textured objects. Recent high-performance general-purpose processors can achieve real-time object detection at a heavy computational cost. However, the processor requires high power consumption and is therefore unsuitable for mobile systems under limited battery conditions. Consequently, a low-power and high-performance HOG feature extraction processor is necessary to widen the range of applications. Figure 1 presents the image resolution versus frame rate for several published descriptions of HOG hardware.

HOG features are adaptable to widely various applications. Consequently, next-generation HOG feature extraction processors must provide higher expandability and higher performance. Therefore,

our goal is to develop design techniques for a real-time HOG feature extraction processor for HDTV resolution video. Parallelized architectures for cell histogram generation, histogram normalization, and SVM classification to reduce the necessary cycle count.

## 2.4 OBSERVATION FROM LITERATURE SURVEY

The advantages and the disadvantages that has been observed from the literature survey along with the methods are given as follow:

1. In [2], Propagative Hough voting is used to provide fast and effective local interest point matching for human activity recognition and detection.
  - Leverage the underlying data distribution of local features to enhance feature matching even when the training data is limited.
  - Ignores the spatio (or spatio-temporal) configuration of local feature points.
  - Used only for classification purpose on the segmented videos without the ability to identify the actions.
1. In [1], A network-based algorithm is proposed for human activity recognition in videos where the entire scene is divided into patches with nodes(network).
  - It can handle scene related as well as group activities efficiently.
  - However, it cannot explicitly differentiate activities with motion patterns in common (e.g., differentiating moving-back-and-forth from moving-forward and moving-backward).
  - Detection process is time consuming due to division of block into patches.

1. In [4], they explicitly model the sequential aspect of activities using a two layer SVM-HCRF with sliding window technique.
  - It uses two-layer SVM hidden conditional random field (SVM-HCRF) rather than large BoW representations.
  - However, it focuses only on sequential modelling of actions rather than action recognition.
  - It also requires huge amount of continuous visual data.
1. In [5], Understanding of the sparse representation and preserves the manifold structure with a generalized version of Laplacian regularized sparse coding for human activity recognition.
  - it captures high semantics and simultaneously finds sparse coordinates for a term in the dictionary.
  - However, this suffers from poor generalization.
  - It cannot perform Multiview learning and deep architecture performance.
1. In [3], FuzzySR algorithm along with BoW to perform continuous human activity segmentation and recognition.
  - It can do segmenting and recognizing a sequence of human activities from continuous unsegmented visual data.
  - It is very sensitive to noise in the time series.
  - The optimization problem due to large size of training data

## **CHAPTER 3**

### **REQUIREMENTS ANALYSIS**

#### **3.1 FUNCTIONAL REQUIREMENTS**

The output of the system is a pop up that shows the action performed by the individual in the input video. The output of the system should adhere to the following requirements.

- The system should be able to clearly differentiate between the background and the foreground.
- The system should be able to identify the action performed in the input video using the hog feature extraction.
- The system should be able to correctly identify the activity performed by the individual in the input video.
- The system must correctly differentiate between the five actions taken into consideration.
- The system must not take too long to identify the action performed by the individual in the input video.

#### **3.2 NON-FUNCTIONAL REQUIREMENTS**

##### **3.2.1 User Interface**

The Users must be able to easily give a video with the specified actions performed as the input. For the given input video, the action is identified. Then, finally a pop up window, which shows activity performed by the individual, will be the overall output of the system.



### **3.2.2 Software**

- Operating System: Windows 8.1
- Programming Language: Matlab
- Software used: Matlab R2015a

### **3.2.3 Performance**

The system must be optimised with high recognition rate, reliable, consistent and should be able to classify the activity with high accuracy.

## **3.3 CONSTRAINTS AND ASSUMPTIONS**

### **Constraints**

- The system would be able to handle only one action at a time. A input video with continuous sequence of actions is not handled in the proposed system.
- The recognition rate of the classifier depends on the extracted HOG features. Changes in the background can adversely affect the output.
- The system cant handle concurrent activity with more than one individual.
- Human interaction is not recognized.

### **Assumptions**

- The input video is assumed to have exactly one individual whose action needs to be identified.
- The input video is assumed to have only action that needs to be identified.
- The input is assumed to be video with a short running time.

# **CHAPTER 4**

## **SYSTEM DESIGN**

### **4.1 PROPOSED SYSTEM**

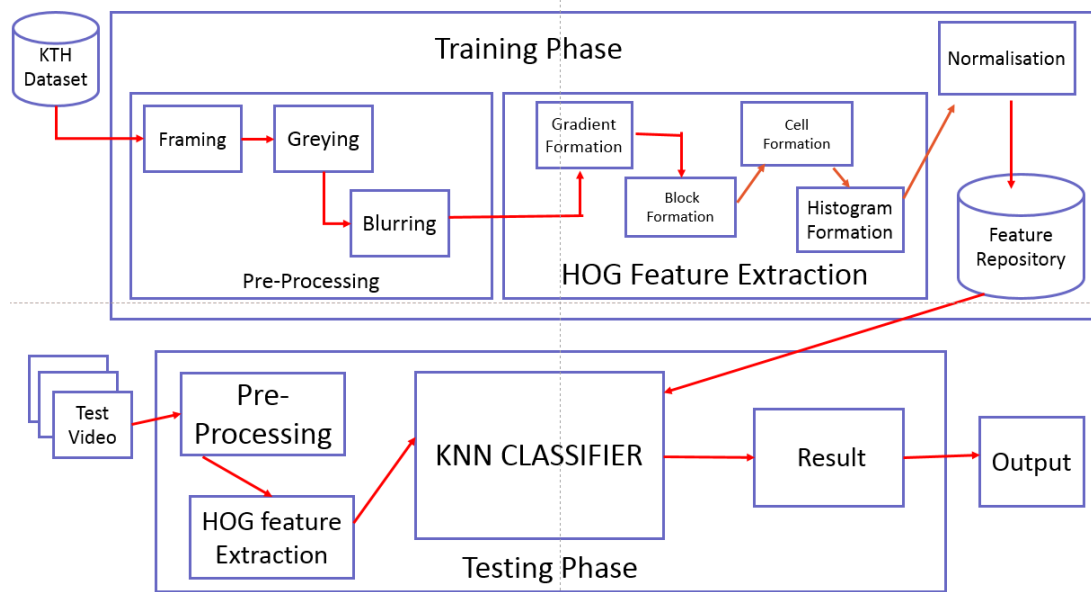
An activity is defined as a maximum continuous period of time during which the activity label is consistent. The system proposed does activity recognition in a continuous visual data that includes complex human activities with no set temporal boundaries. This is done using the concept of KNN Classifier to divide the unsegmented data into meaningful blocks and recognise the human event after the usual video preprocessing (framing, greying and blurring). HOG features are extracted and the feature vectors are used to train a model which is used by the KNN classifier.

Some of the issues addressed are as follows.

1. To bridge the gap between temporal activity segmentation and the BoW representation based on HOG features.
2. To deal with transition effect on temporally adjacent activities.
3. Using KNN Classifier to identify human activities.

### **4.2 SYSTEM ARCHITECTURE**

The system has been divided into seven modules. HOG features are extracted from the KTH dataset. The feature vectors are stored in the database which is used to train a model. The feature vectors are grouped into class labels which is then used by the KNN Classifier to classify the



**Figure 4.1** System Architecture

testing video to the nearest possible label in order to identify the correct activity.

### 4.3 LIST OF MODULES

#### 1. PRE PROCESSING

- Framing of video.
- Greying.
- Blurring.

#### 2. HOG FEATURE EXTRACTION.

- Luminos gradient calculations.
- Bi-linear interpolation.
- Histogram formation.
- Normalisation.

#### 3. ACTION RECOGNITION

- KNN Classifier

## 4.4 SYSTEM MODULES

The System Models [ UseCase Diagram and Sequence Diagram ] for Human Activity recognition is discussed in this section.

### 4.4.1 UseCase Diagram

The user has to give the video that has the human activity. Then, from this input video frames are extracted, from which background is subtracted to get the human image of each frame. These frames undergo greying in order to convert the RGB values into grey scale and blurring. HOG features are then extracted and stored in the feature repository. After class labelling, using the KNN Classifier, human activity is recognised.

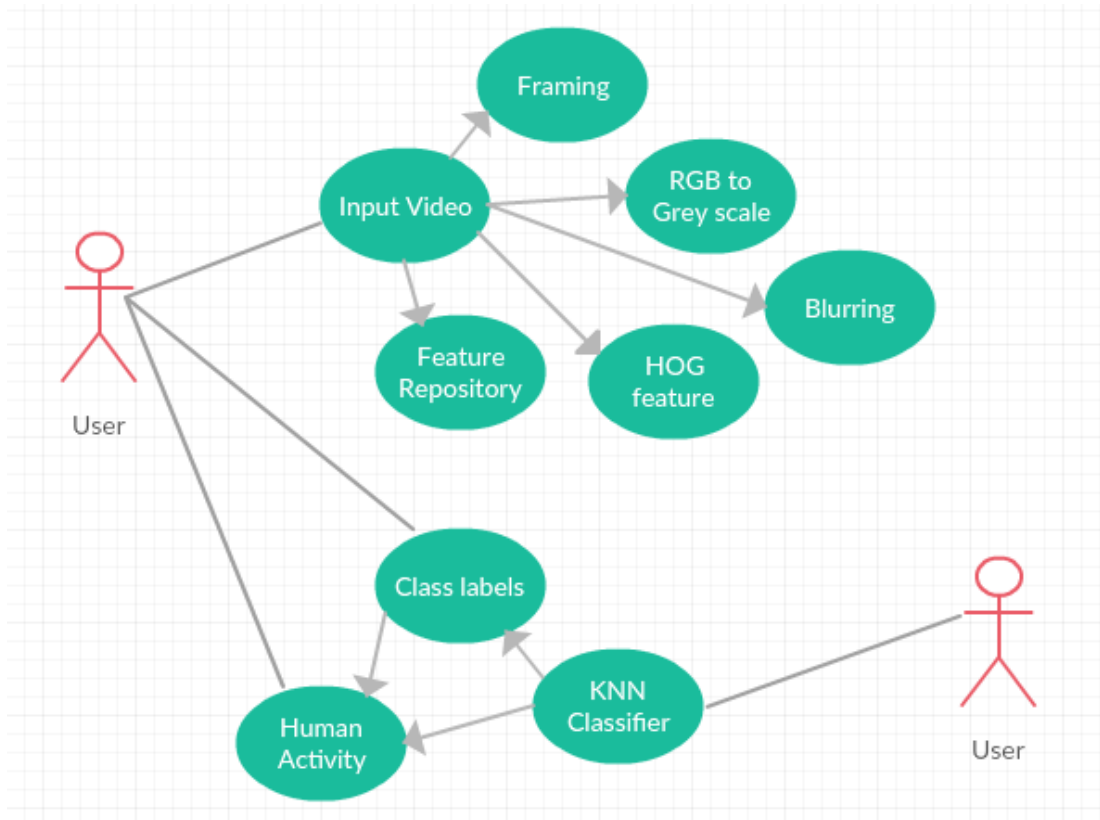
Fig 4.2 represents the usecase diagram for Human Activity identification. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. If  $k = 1$ , then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

**Precondition:** The input video having a single person performing an activity.

**Postcondition:** The output consists of video with the activity label of the activity performed by the person.

### 4.4.2 Sequential diagram

The sequence diagram shows the overall flow of the entire system. The activation regions are given for each module, which denotes the working flow of the modules. The sequences of steps involved in each



**Figure 4.2** UseCase Diagram

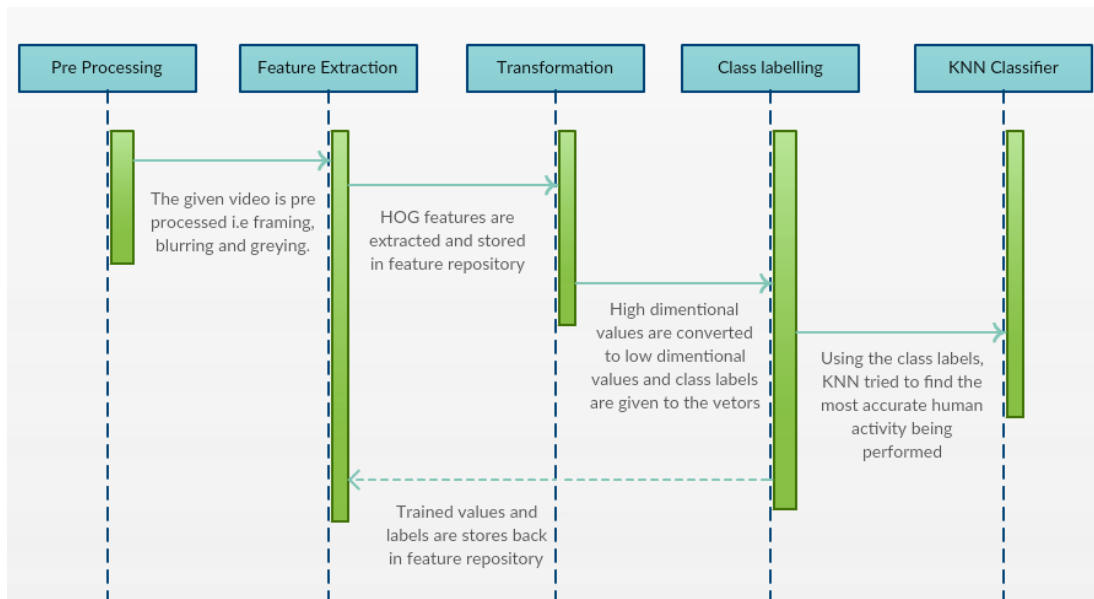
module and between the modules are represented in the figure. Fig 4.3, represents the Sequence diagram for Human Activity Recognition from a video.

## 4.5 MODULES

The detailed description and working of each module is discussed below.

### 4.5.1 Preprocessing

The KTH Dataset is given as input to the preprocessing module. As shown in the figure, framing, RGB to Grey scale conversion and Blurring is done. Fig 4.4, Fig 4.5 represent the entire flow of preprocessing.



**Figure 4.3** Sequence Diagram



**Figure 4.4** Framing

## Framing

Framing is the basic and foremost step involved in video processing. In framing, each frames in the given input video is extracted individually. The frame rate of each video differs, so based on the frame rate, the number of frames in the video changes. The key frames are the frames where the actual activity takes places. It is better to identify these key frames and extract them, instead of extracting entire frames of the given input video. For a short video sequence, the entire frame can be extracted.

The video with the human activity is given as input to this module.



**Figure 4.5** Pre processing

The individual frames are extracted and stored as separate images in the same sequential order as it is extracted. These frame images are given as input for the greying process.

### **Greying**

In this process of Greying, the main aim is to convert higher dimensions to lower dimensions. Since RGB is a 3D data, in order to extract HOG features to process the system, grey scale conversion is done where it is converted into a 2D data.

#### **4.5.2 Gradient Calculation**

An image gradient is a directional change in the intensity or color in an image. The gradient of a two-variable function at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions i.e. in the X and Y direction [ly, lx].

$$Ly_i = im_i - im_{i+2}; \quad (4.1)$$

where,  $i$  is the row of pixel values in each frame  $im$  is  $256*256$  matrix of pixel values of a frame.

$$Lx_j = im_j - im_{j+2}; \quad (4.2)$$

where,  $j$  is the coloumn of pixel values in each frame  $im$  is  $256*256$  matrix of pixel values of a frame.

At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction.

#### 4.5.3 Block and Cell formation

After the Gradient of each image is calculated, each and every image from the video is taken and undergoes the process of block and cell formation. Firstly, each image is divided into  $16*16$  blocks. Each block is further divided into cells of size  $8*8$ .

From each cell, the pixel value of the image is calculated which would be used in the HOG feature extraction formula in order to obtain the feature vectors.

#### 4.5.4 Histogram Formation

Once the pixel value from the previous module is calculated, using the already existing HOG feature formula,

$$H_x = H_x + mag_{p,q} * (j - \alpha)/20; \quad (4.3)$$

$$H_{x+1} = H_{x+1} + mag_{p,q} * (\alpha - i)/20; \quad (4.4)$$

where  $p$   $q$  are the rows and columns and their magnitude denotes the pixel values of patches.  $i$  and  $j$  are the minimum and maximum bin



values respectively.

$$Alpha = anglevalue_{p,q}; \quad (4.5)$$

. Using this formula, the final feature vector is calculated. These feature vectors are eventually stored in a feature repository (database) for further processing by the KNN Classifier. After storing, class labelling takes place in order to group the features together as one activity to enable KNN to correctly and accurately identify the human activity that is being performed in the video.

#### 4.5.5 Normalisation

Normalisation is usually done just to make dealing with feature vectors much easier. A certain max vector is chosen and the any vector that falls beyond that is rejected as it wouldn't have any effect in the identification process.

LI-NORM :

$$S = \sum_0^n |y_i - f(x_i)| \quad (4.6)$$

where  $y_i$  is the target value and  $f(x_i)$  is the estimated value.  $S$  is the minimised sum of the absolute differences.

L2-NORM:

$$S = \sum_0^n (y_i - f(x_i))^2 \quad (4.7)$$

Here,  $S$  gives the minimised sum of the square of the differences between the target and estimated value.

#### 4.5.6 KNN Classifier

Class labelling is done after the feature repository is created. Now using the concept of KNN Classifier, during the testing phase, the HOG

features of the test video is extracted and goes through the KNN algorithm to identify the activity.

The KNN Classifier uses the k-nearest neighbours algorithm. The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If  $k = 1$ , then the object is simply assigned to the class of that single nearest neighbor.

In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors. In the testing phase, once again the HOG features are extracted and processed through KNN. It will try to find the most accurate and closest label from the already trained model in our system and eventually predicting the activity that is being performed.

# **CHAPTER 5**

## **SYSTEM DEVELOPMENT**

### **5.1 SYSTEM FLOW**

The video which is the input given undergoes many stages of pre-processing and feature extraction resulting in extraction of Histogram Oriented Gradients. The type of videos considered are:

1. Boxing video (KTH dataset)
2. Jogging video(KTH dataset)
3. walking video (KTH dataset)
4. hand waving video(KTH dataset)
5. hand clapping video (KTH dataset)

The system is trained by feeding in the KTH dataset with videos of actions listed above.

The database is created which helps in identifying the test footage. In the testing phase, the output from the pre-processing which is a low dimensional value(grayscale) is given to HOG feature extraction module. The output of the HOG feature extraction module which is a normalized histogram based value for each pixel is given to the KNN classifier module which in turn results in identifying human action.

### **5.2 ALGORITHMS**

#### **5.2.1 Preprocessing**

Pre-processing consists of

1. Framing.



**Figure 5.1** Framming of videos

## 2. Greying.

**Input:** Input to the preprocessing module is a set of videos.

**Output:** Grey scaled frames of the video.

**Process:**

In Preprocessing, the input video is segmented into multiple frames, and each of those images are being stored which is later processed in order to calculate the height and width of each image. Fig 5.1 shows the process of framming. Using the calculated data and the RGB value extracted, it is converted to a two dimensional value i.e. greying, inorder to calculate the HOG features. Fig 5.2 and Fig 5.3 shows the calculation of features in two dimensions, X AND Y.

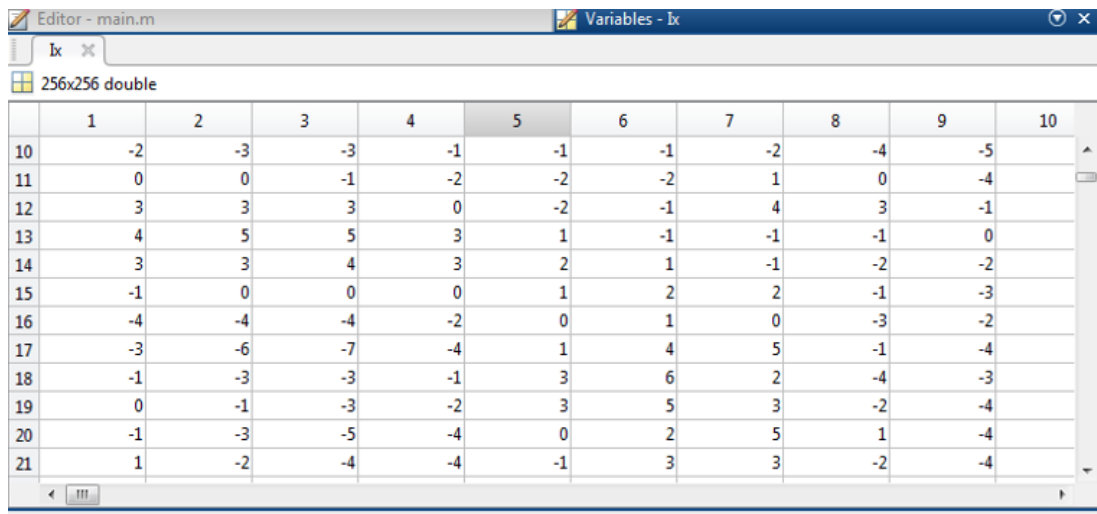
### 5.2.2 HOG Feature Extraction

**Input:** Grey Scaled image frames of each video.

**Output:** Final feature vector stored in a  $50 \times 1729800$  matrix.

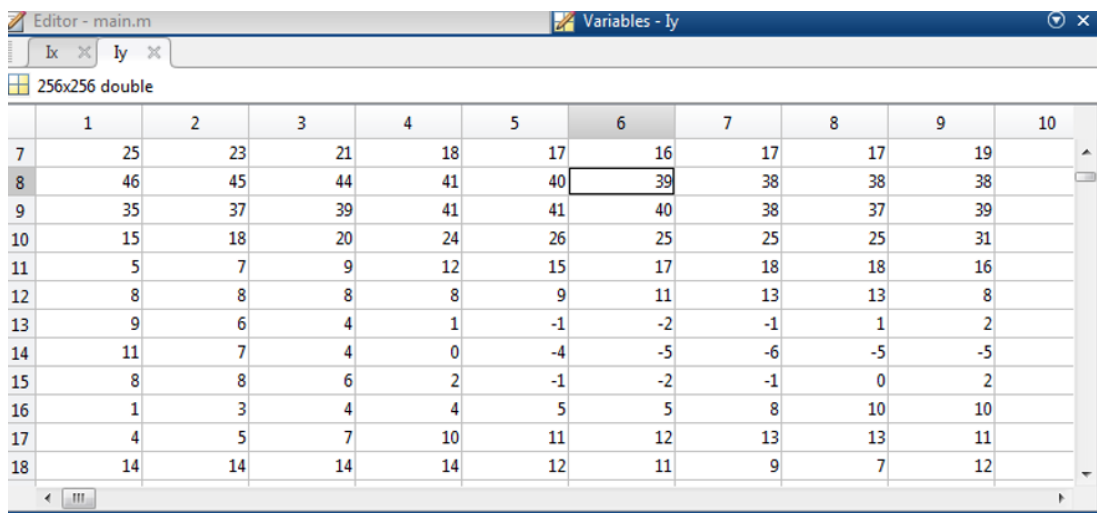
**Process:**

In Algorithm 1, after conversion from three dimension to two, each and every image is divided into blocks of size  $16 \times 16$  and the magnitude



	1	2	3	4	5	6	7	8	9	10
10	-2	-3	-3	-1	-1	-1	-2	-4	-5	
11	0	0	-1	-2	-2	-2	1	0	-4	
12	3	3	3	0	-2	-1	4	3	-1	
13	4	5	5	3	1	-1	-1	-1	0	
14	3	3	4	3	2	1	-1	-2	-2	
15	-1	0	0	0	1	2	2	-1	-3	
16	-4	-4	-4	-2	0	1	0	-3	-2	
17	-3	-6	-7	-4	1	4	5	-1	-4	
18	-1	-3	-3	-1	3	6	2	-4	-3	
19	0	-1	-3	-2	3	5	3	-2	-4	
20	-1	-3	-5	-4	0	2	5	1	-4	
21	1	-2	-4	-4	-1	3	3	-2	-4	

**Figure 5.2** Gradient Calculation in Y direction



	1	2	3	4	5	6	7	8	9	10
7	25	23	21	18	17	16	17	17	19	
8	46	45	44	41	40	39	38	38	38	
9	35	37	39	41	41	40	38	37	39	
10	15	18	20	24	26	25	25	25	31	
11	5	7	9	12	15	17	18	18	16	
12	8	8	8	8	9	11	13	13	8	
13	9	6	4	1	-1	-2	-1	1	2	
14	11	7	4	0	-4	-5	-6	-5	-5	
15	8	8	6	2	-1	-2	-1	0	2	
16	1	3	4	4	5	5	8	10	10	
17	4	5	7	10	11	12	13	13	11	
18	14	14	14	14	12	11	9	7	12	

**Figure 5.3** Gradient Calculation in X direction

and angle of each patches is calculated. Similarly, we traverse through each and every block and is divided into cells of size 8x8 and the magnitude and angle of A using the patches is calculated. Using the calculated patches and angel, Aphla is calculated. Fig 5.4 and Fig 5.5 shows the formations of blocks and then cells.

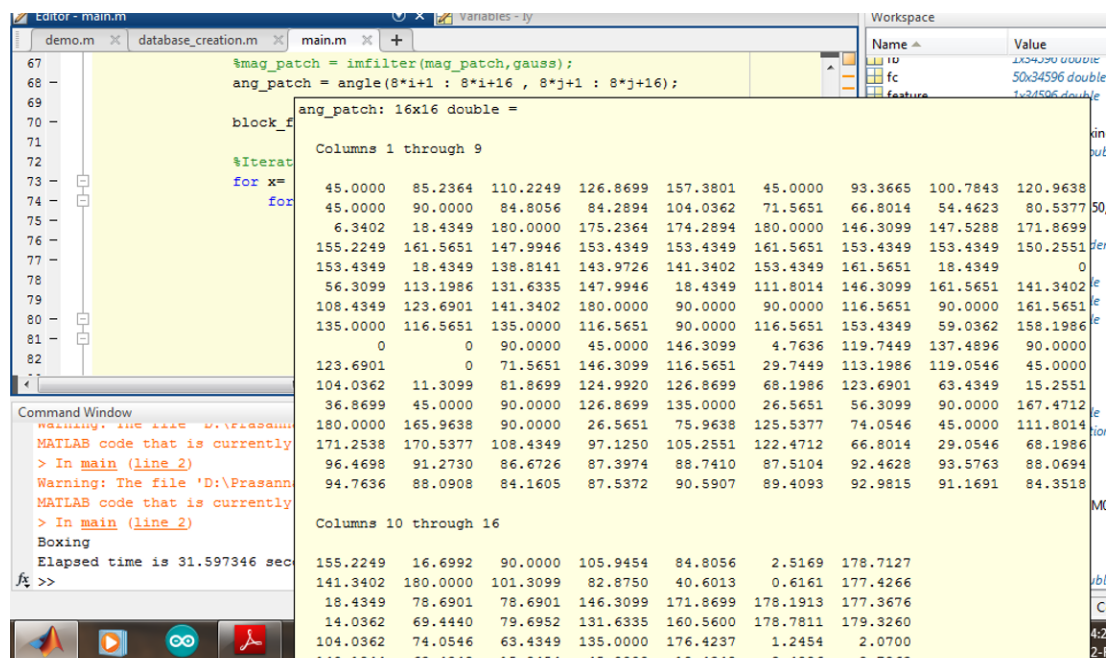
In Algorithm 2, using the calculated value of alpha, and if-else-if ladder is created and is applied to the histogram formula.

---

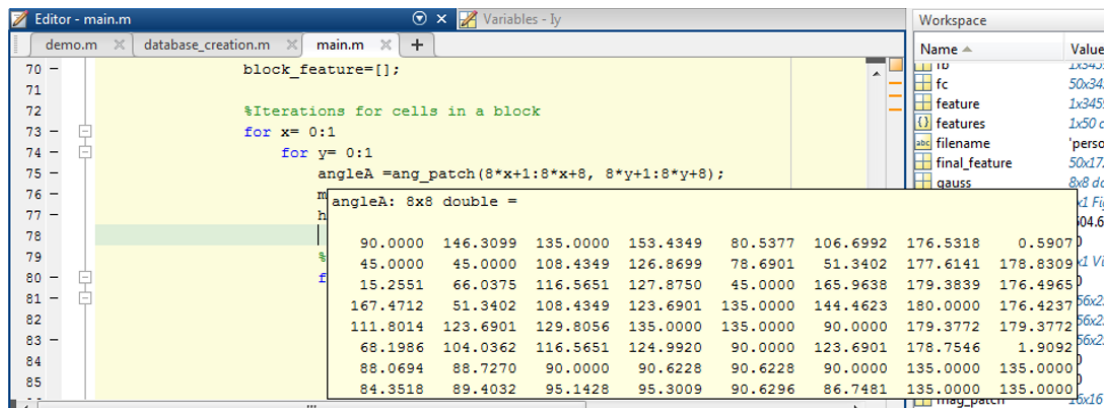
**Algorithm 1** Iteration of Blocks and Cells
 

---

1. for  $i=0$  to  $\text{row}/8$
  2. for  $j=0$  to  $\text{cols}/8$
  3. calculate  $\text{magp}$ ,  $\text{angp}$ ; (divides the frame into  $16 \times 16$  blocks)
  4. for  $x=0$  to 1
  5. for  $y=0$  to 1
  6. calculate  $\text{angA}$ ,  $\text{magA}$ ; (divides a block into  $8 \times 8$  cells)
  7. for  $p=1$  to 8
  8. for  $q=1$  to 8
  9. calculate  $\alpha$ ;
- 



**Figure 5.4** Iteration of blocks.



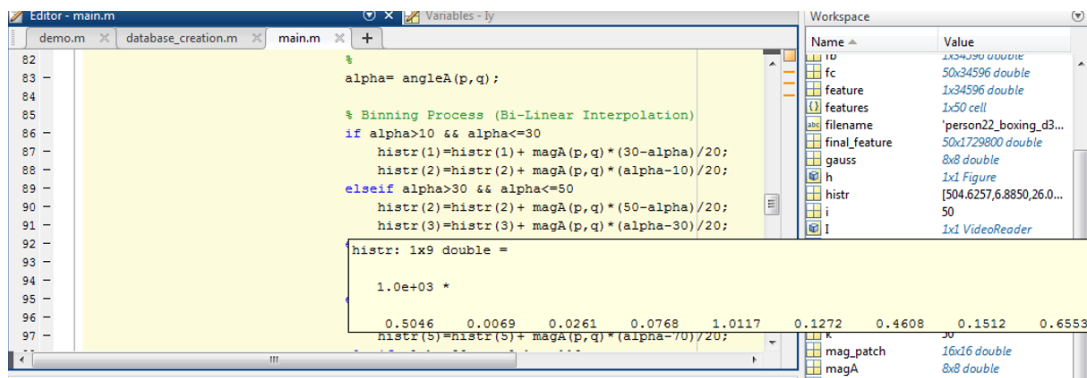
**Figure 5.5** Iteration of cells.

---

### Algorithm 2 Histogram Formation

---

1. If alpha GT 10 and alpha LT= 30
  2. hist(1) and hist(2) is calculated
  3. If alpha GT 30 and alpha LT = 50
  4. hist(2) and hist(3) is calculated
  5. If alpha GT 50 and alpha LT= 70
  6. hist(3) and hist(4) is calculated
  7. If alpha GT 70 and alpha LT= 90
  8. hist(4) and hist(5) is calculated
  9. If alpha GT 90 and alpha LT= 110
  10. hist(5) and hist(6) is calculated
  11. If alpha GT 110 and alpha LT= 130
  12. hist(6) and hist(7) is calculated
  13. If alpha GT 130 and alpha LT= 150
  14. hist(7) and hist(8) is calculated
  15. If alpha GT 150 and alpha LT= 170
  16. hist(8) and hist(9) is calculated
  17. If alpha GT= 0 and alpha LT= 0
  18. hist(1) and hist(9) is calculated
  19. If alpha GT 170 and alpha LT= 180
  20. hist(9) and hist(1) is calculated
-



**Figure 5.6** Histogram Formation.

### 5.2.3 Activity Detection

This module uses the KNN classifier and the class labels to identify the action performed in the input video. The stored training phase database is used for action recognition. K-nearest neighbour algorithm. will choose the most accurate activity value from database. Pop ups are used to display the correct identified action.

**Input:** Feature vectors.

**Output:** Pop up window showing the action performed in the video.

**Process:**

In Algorithm 3, using the feature vector of the test video and the stored and labelled feature vector, KNN is used to find the closest neighbouring match to identify the action being performed. The Euclidean distance is calculated initially between the final and test feature vectors. The calculated points are then arranged in an ascending order. Taking the first few points and trying to match with the nearest neighbours to find its class label association and then, the activity is identified.

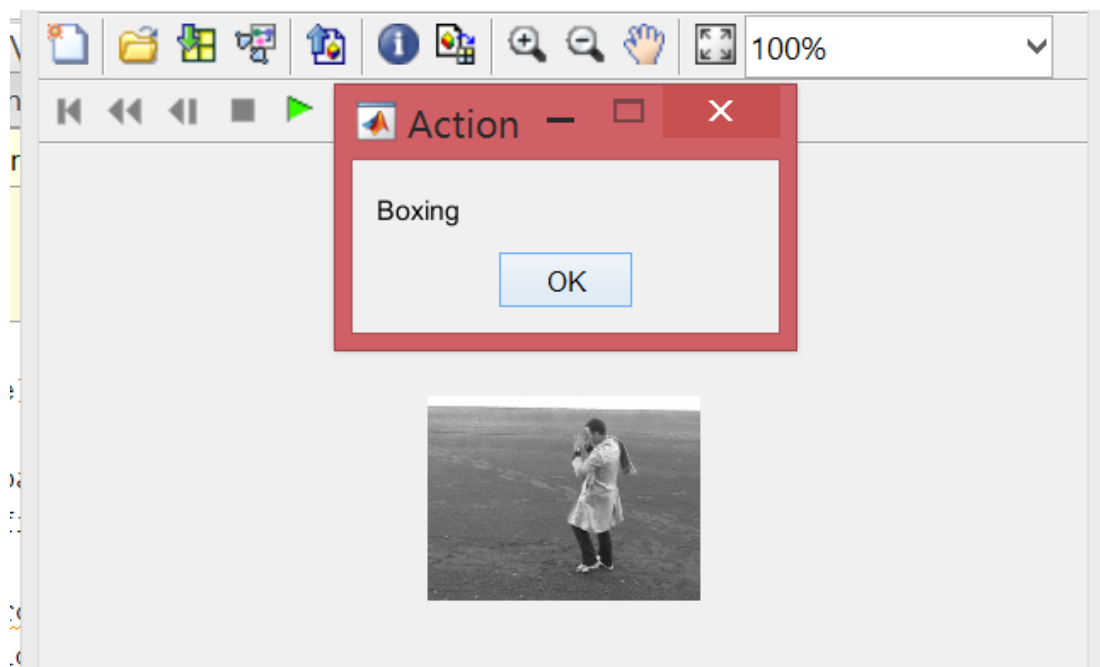


---

**Algorithm 3** K-nearest neighbour
 

---

1. Calculate  $d(x, x_i)$   $i = 1, 2, \dots, n$ ; where  $d$  denotes the Euclidean distance between the points.
  2. Arrange the calculated  $n$  Euclidean distances in non-decreasing order.
  3. Let  $k$  be a +ve integer, take the first  $k$  distances from this sorted list.
  4. Find those  $k$ -points corresponding to these  $k$ -distances.
  5. Let  $k_i$  denotes the number of points belonging to the  $i$ th class among  $k$  points i.e.  $k \geq 0$
  6. If  $k_i \geq k_j$   $\forall j \neq i$  then put  $x$  in class  $i$ .
- 



**Figure 5.7** Final Result.

## **CHAPTER 6**

### **RESULTS AND DISCUSSION**

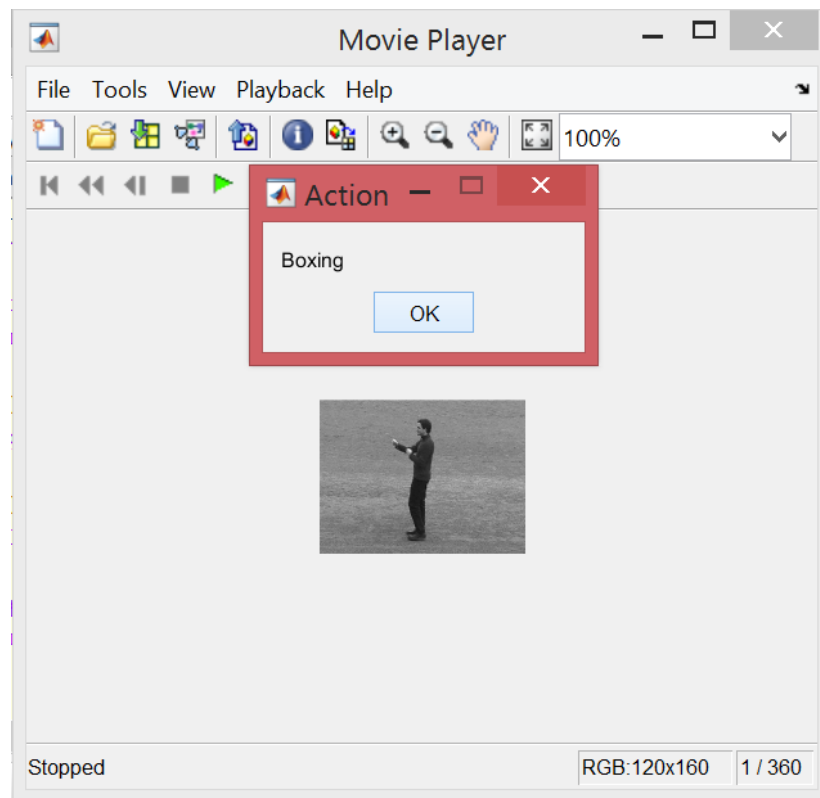
The Results obtained from the system has been discussed in the following subsections. The performance of the system has been evaluated using four metrics Accuracy, Recall, Precision and F score. These four metrics have been represented graphically to provide a pictorial representation of the results. Furthermore, time centric performance metrics are also taken into consideration. The average training time for each action set is calculated and is represented graphically. The average time taken for the identification of the action during the testing phase is also represented graphically. The four performance metrics are also represented in a tabular format.

#### **6.1 DATASET**

KTH dataset is a collection of short videos depicting 5 actions by an individual. The five actions are : Boxing, hand waving, hand clapping, jogging and walking.

#### **6.2 IMPLEMENTATION DETAILS AND DISCUSSION**

In this section the implementation details of all the actions are shown using a snapshot of the result. The results can be either boxing, handwaving, handclapping, jogging or walking.



**Figure 6.1** Recognised Boxing.

### **Boxing**

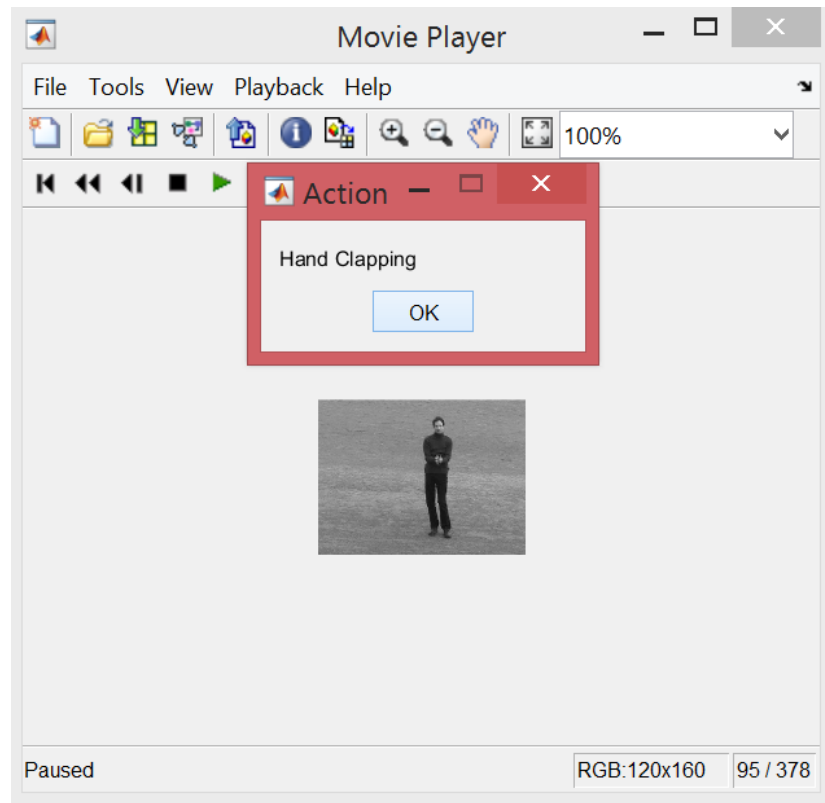
The figure 6.1 shows the snapshot of boxing The accuracy of identification is 70

### **Hand clapping**

The figure 6.2 shows the snapshot of boxing The accuracy of identification is 80

### **Hand waving**

The figure 6.3 shows the snapshot of boxing The accuracy of identification is 70



**Figure 6.2** Recognised hand clapping.

## Jogging

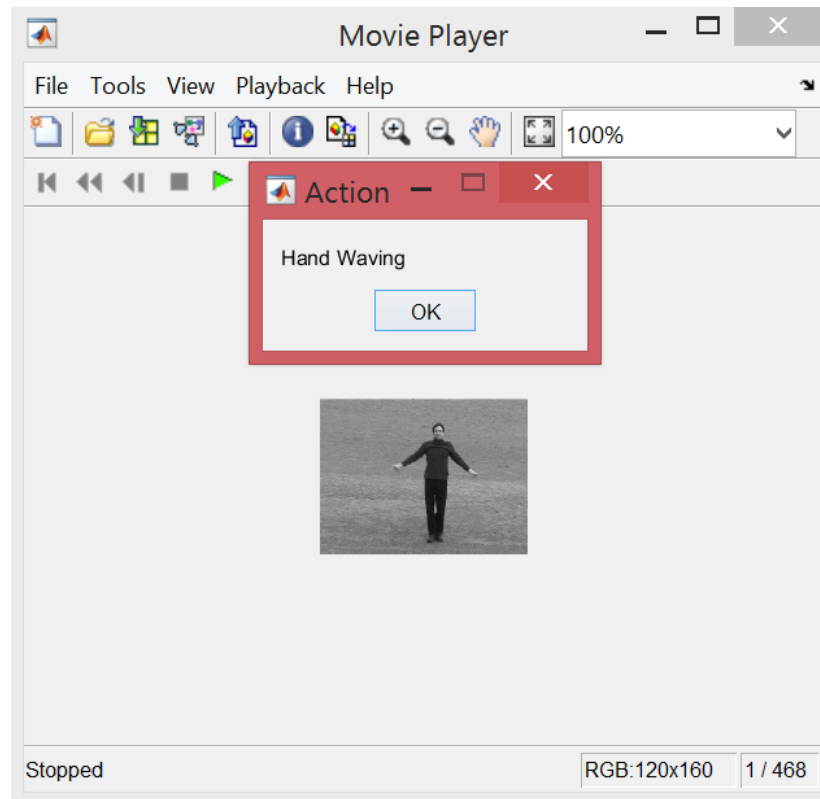
The figure 6.4 shows the snapshot of boxing The accuracy of identification is 100

## walking

The figure 6.5 shows the snapshot of boxing The accuracy of identification is 80

### 6.2.1 Evaluation

The recognition rate of different actions by the classifier for the given transformation is found out and the corresponding graph is plotted, which shows that our proposed algorithm is more effective compared to



**Figure 6.3** Recognised hand waving.

other existing transformations.

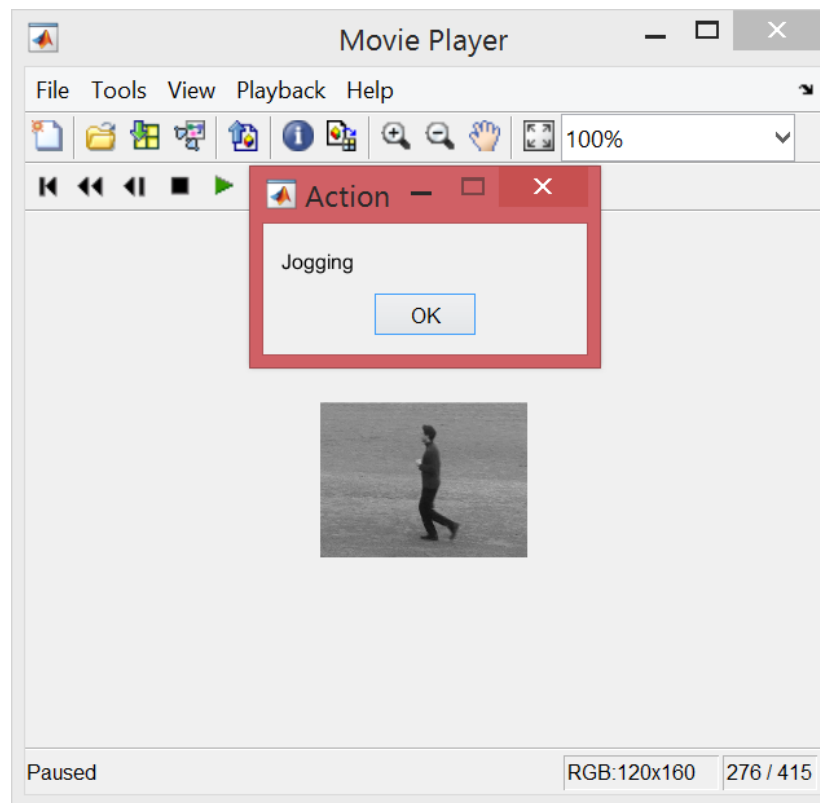
$$Precision = \frac{(Number\ of\ interaction\ identified\ correctly)}{(Number\ of\ interaction\ of\ positive\ recognition\ found)} \quad (6.1)$$

$$Recall = \frac{(Number\ of\ interaction\ of\ positive\ recognition\ found)}{(Total\ number\ of\ relevant\ input\ interaction)} \quad (6.2)$$

$$F - score = \frac{2(precision\ recall)}{(precision + recall)} \quad (6.3)$$

**Training time** = Amount of time spent to train the system.

**Recognition time** = Time taken by the system to recognize the activity.

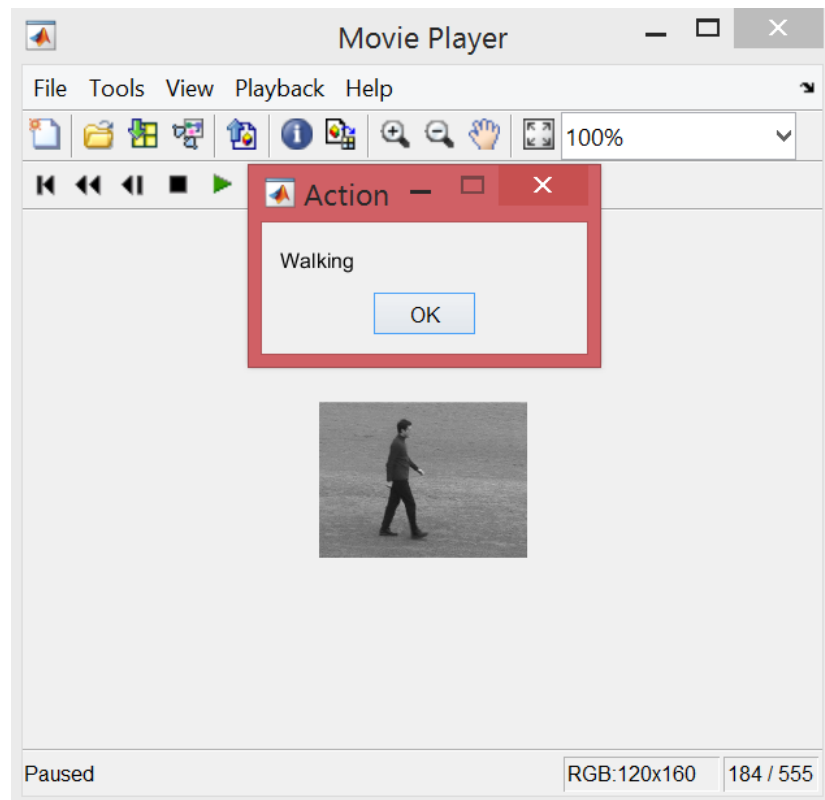


**Figure 6.4** Recognised jogging.

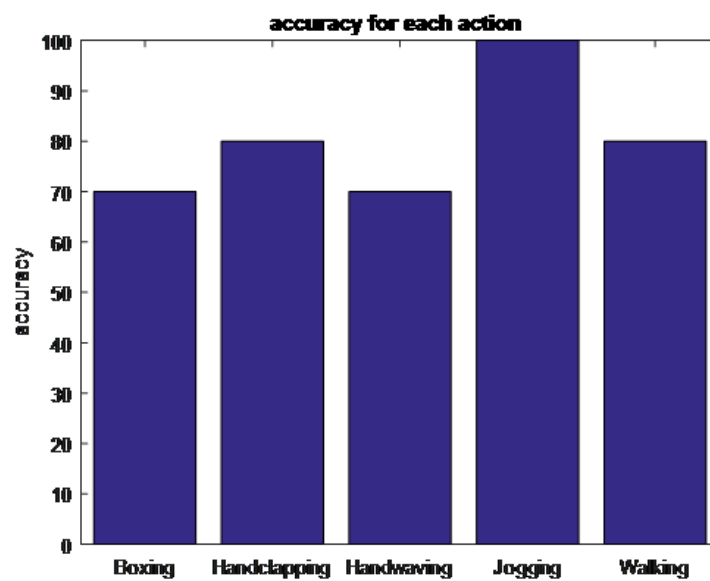
Fig 6.6 indicates the Accuracy graph indicating how accurately the system identifies the human activity from the video. From the graph we can infer that the jogging accuracy is almost 1. While Hand clapping and walking are at 0.8 and boxing and hand waving at 0.7

Fig 6.7 indicates a combined graph of precision, recall and f-score of all the test videos. It can be inferred from the graph that the precision is 1 for hand clapping, hand waving, jogging and walking. The system is able to identify jogging correctly for every relevant input video.

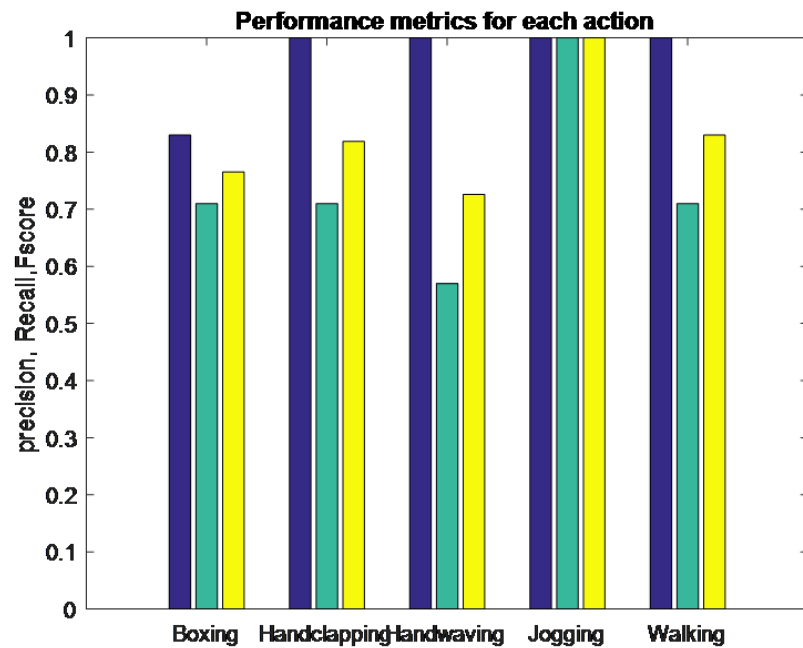
Fig 6.8 indicates the training time required for various human activities performed in each video. It can be inferred from the graph that training the system for the activity walking takes 46 seconds making it the slowest activity to train.



**Figure 6.5** Recognised walking.



**Figure 6.6** Accuracy Graph.



**Figure 6.7** Precision, Recall and Fscore Graph.

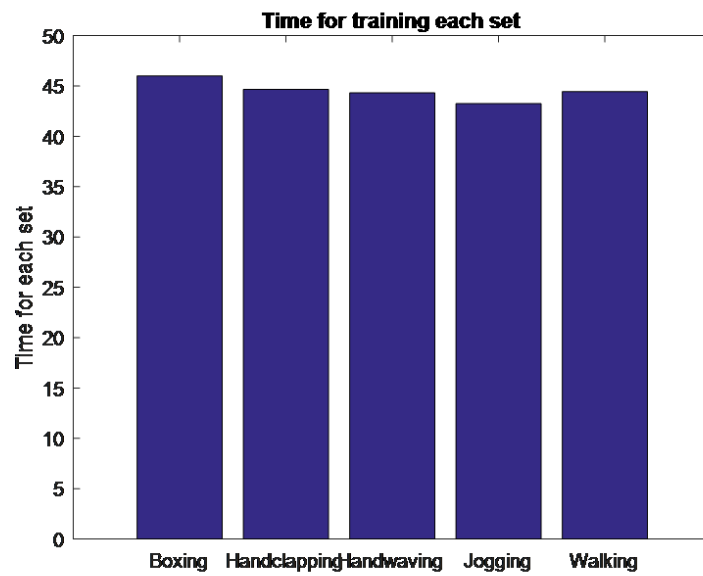
S.No	Activity	Accuracy	Precision	Recall	f-score
1	Boxing	0.70	0.83	0.71	0.76
2	Hand Clapping	0.80	1	0.71	0.81
3	Hand Waving	0.70	1	0.71	0.81
4	Jogging	1	1	0.57	0.72
5	Walking	0.80	1	0.71	0.83

**Table 6.1** Performance Matrix.

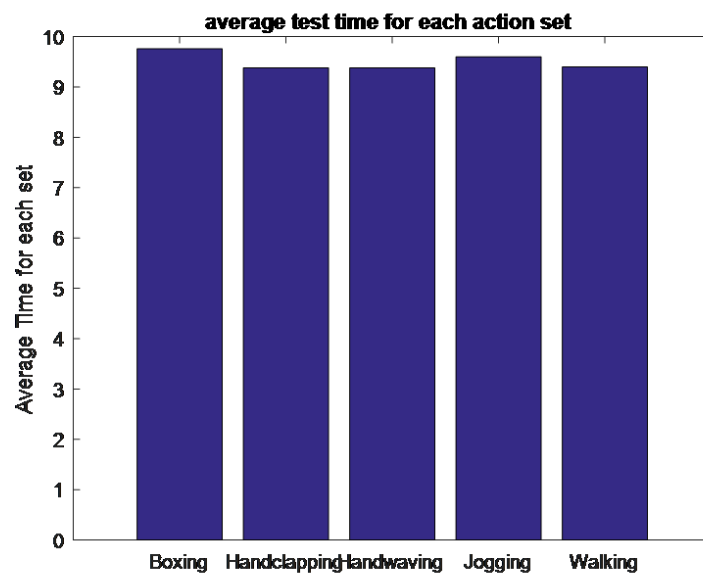
S.No	Activity	Training Time(secs)	Testing Time(secs)
1	Boxing	44.31	9.76
2	Hand Clapping	43.24	9.38
3	Hand Waving	44.45	9.38
4	Jogging	44.65	9.60
5	Walking	46.00	9.40

**Table 6.2** Training and Testing time.





**Figure 6.8** Training Graph.



**Figure 6.9** Testing Graph.

S.no	Activity	Time duration (secs)
1	Boxing	13:00-17:00
2	Hand Clapping	11:00-19:00
3	Hand Waving	18:00-31:00
4	Jogging	12:00-19:00
5	Walking	16:00-27:00

**Table 6.3** Time Duration table.

Fig 6.9 indicates the testing time required to test each videos to identify the correct activity. It can be inferred from the graph that boxing taken the longest amount of time to be identified correctly while hand clapping and hand waving takes the least.

### 6.2.2 Test Cases

The test cases considered here comprises of 10 videos for each action namely boxing, hand clapping, hand waving, jogging and walking. In the 10 videos considered for an action, 7 of them depict the given action and is known to the tester. The other 3 videos are randomly selected that could be videos depicting any other action from the given list of actions. The three random videos are selected to check for false positives and true negatives. The test cases are numbered from TC1-TC10. TC8-TC10 are the videos depicting other actions.

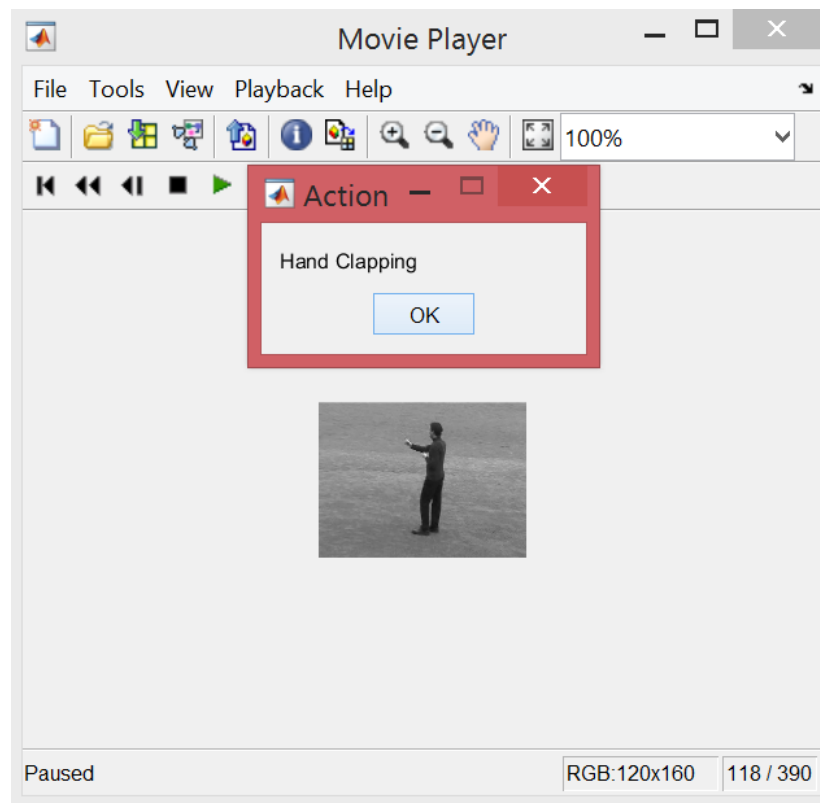
#### Boxing

$$TP = 5$$

$$TN = 2$$

$$FP = 1$$

$$FN = 2$$



**Figure 6.10** Test case for Boxing

In the boxing test cases ,the TC2 gives us a false negative. This means that the video which depicts boxing is being identified as some other action.

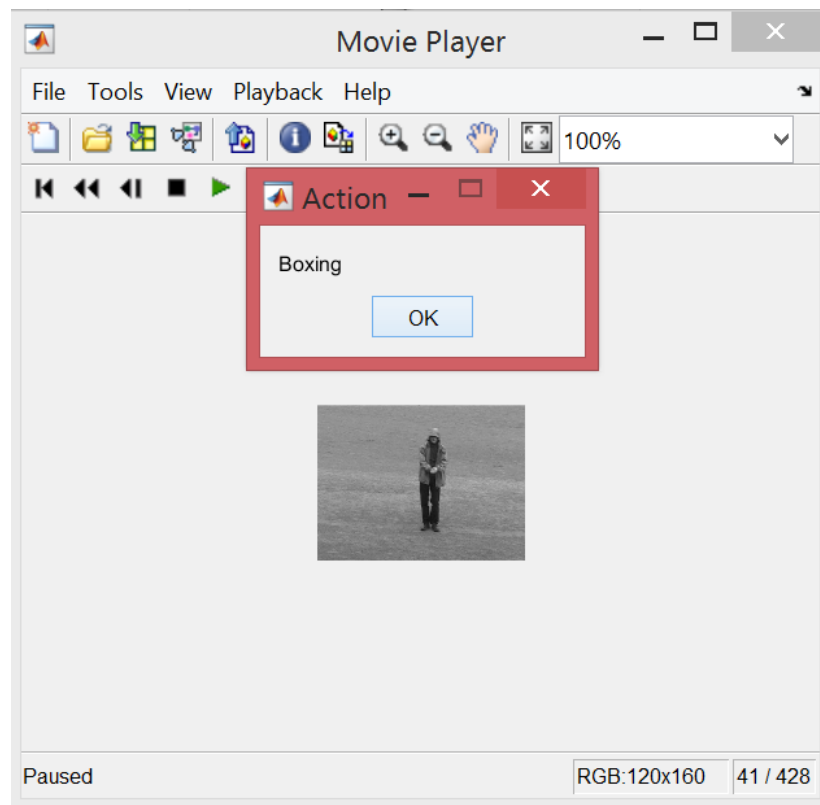
In this case boxing is wrongly identified as hand clapping. From Fig 6.10, is a video that is wrongly identified as hand clapping which results in a false positive. The other 2 videos that depicting actions other than boxing are correctly identified which gives us two true negative values. The rest five values are true positives.

### **Hand Clapping**

$$TP = 5$$

$$TN = 3$$

$$FP = 0$$



**Figure 6.11** Test case for Hand Clapping.

FN = 2

The test cases used for hand clapping does not result in any false positives. That is, it doesn't identify some other action as hand clapping. All the three random action videos are identified correctly which gives us three true negatives. Fig 6.11 indicates that hand clapping video is wrongly identified as boxing.

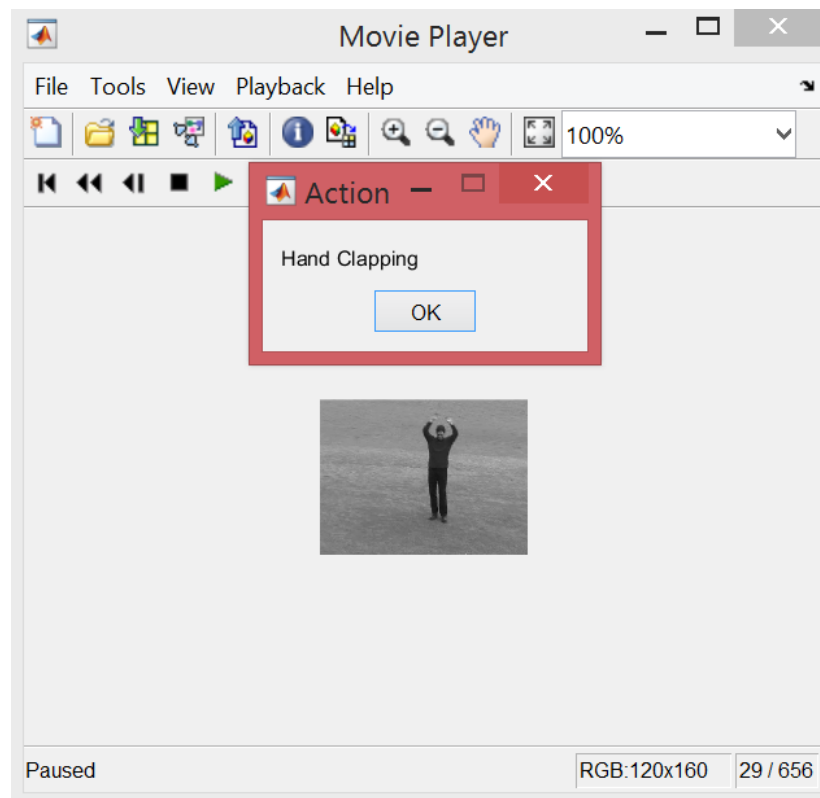
### **Hand Waving**

TP = 4

TN = 3

FP = 0

FN = 3



**Figure 6.12** Test case for Hand Waving.

The hand waving test cases has no false positives also. All the three random action videos are correctly identified which gives us three true negatives. Fig 6.12 indicated that the hand waving video has been wrongly identified as hand clapping.

### **Walking**

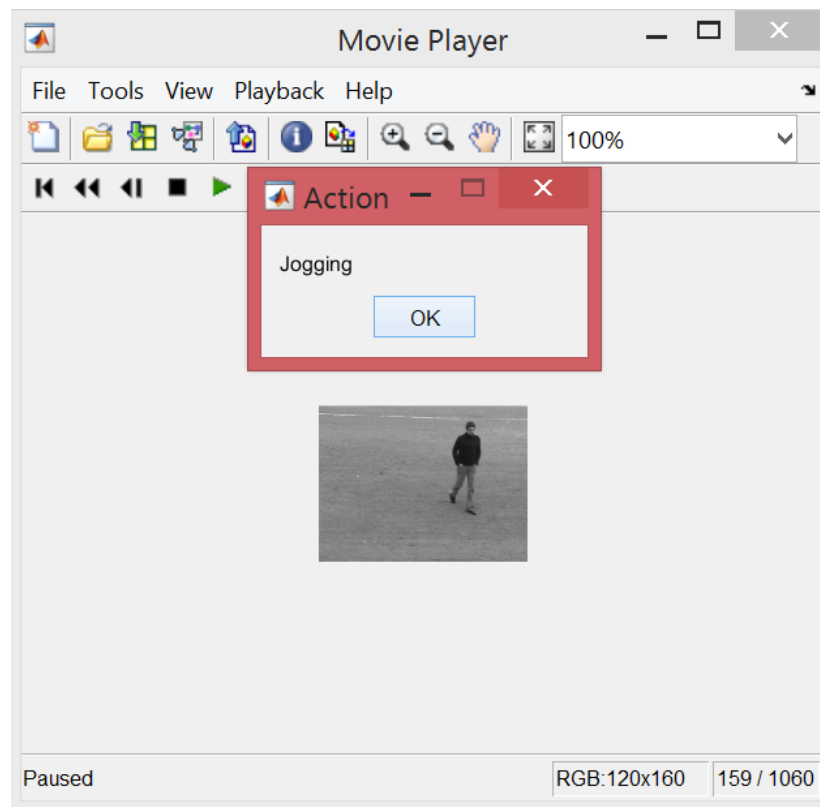
$$TP = 5$$

$$TN = 3$$

$$FP = 0$$

$$FN = 2$$

The test cases used for walking does not result in any false positives. That is, it doesnt identify some other action as hand clapping.



**Figure 6.13** Test case for Walking.

All the three random action videos are identified correctly which gives us three true negatives. Fig 6.13 a video of walking has been wrongly identified as jogging.

### **Jogging**

$$TP = 7$$

$$TN = 3$$

$$FP = 0$$

$$FN = 0$$

In the jogging test cases, all the test cases were correctly identified. The first seven videos depicting jogging were identified as jogging and the next three random action videos were also correctly identified. This

results in no false positives or false negatives.

### **6.2.3 Performance Metrics**

The accuracy for the actions Boxing , Hand clapping, Hand waving, jogging, walking are 70, 80, 70, 100, 80 respectively. The average accuracy of the testing phase for all the five actions is 80. The precision for the five actions listed above are 0.83, 1, 1, 1, 1. The average precision of identification of the five given actions are 0.96. The recall values for the five actions in the same order are 0.71, 0.71, 0.57, 1, 0.71.

The average recall value for all the five actions are 0.74. The f-score values for the five actions are 0.76, 0.81, 0.72, 1, 0.83.

The average f-score value is 0.82.

The system was put through 5 types of testcases, each type for one action. The system was fed 10 videos for each action. The test for one action has 7 videos of that action performed known to the tester and 3 random videos of other actions. This was done to test the true negative results and false positive results.

## **CHAPTER 7**

### **CONCLUSION**

#### **7.1 CONCLUSION**

We have introduced the KNN algorithm to perform continuous human activity segmentation and recognition. Given a video containing continuous human activities, after uniformly partitioning the video into disjoint blocks, our algorithm computes the human activity distribution of each block through mapping high-dimensional discrete feature space to real-valued activity space. Then, the summaries are used to form a feature vector repository which is then labelled in order to train a model which would then be used by the KNN Classifier to identify the activity. Empirical studies are conducted using six real-world human activity datasets, with a focus on temporally segmenting and probabilistically recognizing continuous human daily activities from both color and RGB-D visual data in human social environments.

#### **7.2 FUTURE WORK**

The basic arrangement can be extended to newer areas of application. Human activity recognition can be extended to various domain. The following are some of the future scope of the proposed system.

- Ambient assisted living systems use activity recognition techniques to monitor and assist residents to secure their safety and well-being. Some of the examples are the GERHOME, HERMES project etc.



- Health care monitoring systems are designed based on the combination of one or more activity recognition components such as fall detection, human tracking, security alarm or cognitive assistance components. Most of the health care systems use body-worn and contextual sensors that are placed on patients bodies and in their environment.
- The surveillance system is able to recognize human behavior such as fighting or vandalism events occurring in a metro system by using one or several camera. Additionally, the system was able to detect and predict the suspicious and aggressive behavior of a group of individuals in a prison. They are all based on object-detection method.

## REFERENCES

- [1] Weiyao Lin, Yuanzhe Chen, Jianxin Wu, Hanli Wang, Bin Sheng, and Hongxiang Li ,“A New Network-Based Algorithm for Human Activity Recognition in Videos”, IEEE Transactions On Circuits and Systems for Video Technology Vol. 24, No. 5, May 2014.
- [2] Gang Yu, Junsong Yuan and Zicheng Liu,“Propagative Hough Voting for Human Activity Detection and Recognition”, Journal of Latex Class Files, Vol. 11, No. 4, December 2013.
- [3] Hao Zhang, Wenjun Zhou and Lynne E. Parker, “Fuzzy Temporal Segmentation and Probabilistic Recognition of Continuous Human Daily Activities”, IEEE Transactions On Human-Machine Systems,2168-2291 2015 IEEE.
- [4] Mouna Selmi, Mounm A. El-Yacoubi and Bernadette Dorizzi,“Two-layer discriminative model for human activity recognition”, IEEE Signal Process. Mag.,2016, Vol. 10 Iss. 4, pp. 273-278 .
- [5] Weifeng Liu, Zheng-Jun Zha, Yanjiang Wang, Ke Lu and Dacheng Tao,“p-Laplacian Regularized Sparse Coding for Human Activity Recognition”, IEEE Transactions on Industrial Electronics, 0278-0046 (c) 2015 IEEE.
- [6] A. K. R. Chowdhury and R. Chellappa, “A factoriza-

tion approach for activity recognition”, in Proc. IEEE Conf. Comput. Vision Pattern Recog. Workshop, 2003, pp. 1622.

[7] J.Ben-Arie, Z.Wang, P.Pandit, and S.Rajaram, “Human activity recognition using multidimensional indexing”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 8, pp. 10911104, Aug. 2002.

[8] J. Luo, W. Wang, and H. Qi, “Group sparsity and geometry constrained dictionary learning for action recognition from depthmaps”, in Proc. IEEE Int. Conf. Comput. Vision, 2013, pp. 18091816.

[9] X. Ji and H. Liu, “Advances in view-invariant human motion analysis: A review”, IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev., vol. 40, no. 1, pp. 1324, Jan. 2010.

[10] E. S. Page, “Continuous Inspection Schemes”, Biometrika, vol. 41, pp. 100115, 1954.