Human Activity Recognition in Videos

Syed Waqar Ali
01-134141-125
M. Yousaf Saqib
01-134141-095

Supervisor: Dr. Imran Siddiqi

Department of Computer Science
Bahria University, islamabad.

Contents

1 Introduction **3**
1.1 Objective . . . 3
1.2 Problem Description . . . 4
1.3 Methodology . . . 4
1.4 Project Scope . . . 5
1.5 Organiztion of thesis . . . 5

2 Literature Review **7**
2.1 Identify human activities on high-end phones using a word-friendly vector support . . . 7
2.2 Familiar with activities that accelerate user data annotation . . . 7
2.3 Identify the activity of accelerometer data on the phone . . . 8
2.4 Decision rules based on recognition of human activity . . . 8
2.5 Uniform framework for monitoring and analyzing human body movement . . . 8
2.6 Motion Detection Based on Frame Difference Method . . . 9
2.7 Human Activity Recognition in Videos . . . 9

3 Requirement Specifications **11**
3.1 Functional requirements . . . 11
3.1.1 Processing videos . . . 11
3.1.2 Input Video . . . 11
3.1.3 MHI . . . 11
3.1.4 Feature Detector . . . 11
3.1.5 Classifier . . . 12
3.2 Non Function Requirements . . . 12
3.2.1 Availability . . . 12
3.2.2 Reliable . . . 12
3.2.3 Security . . . 12
3.2.4 Usability . . . 12
3.2.5 User Friendly . . . 13
3.2.6 Use case . . . 13
3.3 Use Case . . . 13
3.3.1 Use Case 1: Start of Application . . . 14

3.3.2 Use Case 2: Uploading a Query Video 15
3.3.3 Use Case 3: Identify Activity 16

4 DESIGN **17**
4.1 System Architecture 17
4.2 Design Methodology 18
4.3 High Level Design 18
4.3.1 Component Diagram 19
4.3.2 Activity Diagram 19
4.4 GUI Design 20

5 System Implementation **25**
5.1 Introduction 25
5.1.1 Algorithmic Workflow 25
5.2 System Components 26
5.2.1 Selection of video 27
5.2.2 Frame Difference 27
5.2.3 Feature Extraction 27
5.2.4 Activity classification 28
5.3 Tools and Technologies 28

6 System Testing and Evaluation **29**
6.1 Motion History Image Creation 29
6.2 HOG Creation 29
6.2.1 Results Summary 33
6.3 Usability Testing 34
6.4 Compatibility Testing 35
6.5 Software Performance Testing 35
6.6 Graphical User Interface Testing 35
6.7 Test Cases 36
6.7.1 Application startup test case 36
6.7.2 Query video loading test case 36
6.7.3 Identify query video test case 36

7 Conclusion & Future Work **39**
7.1 Conclusion 39
7.2 Future Work 39

List of Figures

1.1 Snapshots of activities in the database [4] (a) Jogging (b) Boxing (c) Clapping (d) Waving 4

3.1 Training Use Case 13
3.2 Testing Use Case 14
3.3 Start of Application 14
3.4 Upload a Query Video 15
3.5 Identify Activity 16

4.1 System Architecture 17
4.2 component Diagram 19
4.3 Activity Diagram 19
4.4 Starting Step 20
4.5 Show Dialog Box 21
4.6 Running video on form 21
4.7 Feature Extraction 22
4.8 Final Result 23

5.1 Algorithm Work Flow 26

6.1 Guide Link 34
6.2 Guide Form 35

List of Tables

3.1 Start of Application 15
3.2 Upload a Query Video 15
3.3 Identify Activity 16

6.1 Clapping Outcome 30
6.2 Boxing Outcomes 31
6.3 Waving Outcomes 32
6.4 Jogging Outcomes 33
6.5 Overall Results 33
6.6 Application startup test case 36
6.7 Query video loading test case 36
6.8 Identify query video test case 37

Chapter 1

Introduction

Security and surveillance have gained significant importance during the last decade. There has been an increasing trend towards the installation of surveillance cameras at public places including airports, bus stations, shopping centers and important roads. These cameras send live feed to a control room where the videos are continuously monitored by human operators. With the advancements in different areas of computer vision, attempts have been made to develop automatic surveillance systems capable of recognizing different activities. Research on these automatic surveillance systems, however, is still in its early days as computerized recognition of human actions is a complex and challenging problem.

The key challenges in automatic human activity recognition include segmentation, occlusion and modeling a given activity. In some cases where the behavior is governed by a set of underlying rules (for instance motion analysis of soccer players), models can be generated for each activity of interest [1]. Motion analysis of unconstrained street or crowd scenes, on the other hand, is more challenging to model. Another significant challenge is the level at which the video frames are processed and analyzed. For instance, studying the behavior of an individual with in a street scene may not be very meaningful unless the complete crowd behavior is studied [2]. The present project also targets this interesting area of human activity recognition from videos.

1.1 Objective

The aim of the proposed project is to develop a system that can analyze videos and detect the activity being performed in a video from a set of predefined activities. The development of such a system targets the following objectives.

- Acquaintance with the state-of-the-art techniques for human activity recognition.
- Analysis of videos to segment humans.

- Extracting useful features to characterize different activities.
- Training models on a set of pre-defined activities.
- Classification of given video into one of the trained activities.

1.2 Problem Description

Human activity recognition is an active area of research that has gained significant research attention over the last few years. A number of computer vision research labs are contributing to this emerging area through development of databases, ground truth labeling and design and implementation of activity recognition algorithms [3]. This project aims to develop a system that can analyze given videos and recognize the activity being performed. The activities considered in our study include jogging, clapping, boxing and hand waving. More details on these activities are presented in the next section.

1.3 Methodology

The done methodology for activity recognition relies on a supervised approach. We worked on the benchmark activity recognition database publicly available for research purposes [4]. A snapshot of each of the four activities considered in our study is presented in Figure 1.

Figure 1.1: Snapshots of activities in the database [4] (a) Jogging (b) Boxing (c) Clapping (d) Waving

The available set of videos are divided into training and test sets. For each of the videos in the training set, frames are extracted. Motion history image then extracted characterizing each activity. HOG descriptor is then applied to MHI which gives the gradient points. These points are then used to train the system. After training the testing is done by taking the Euclidian distance form training data and then KNN is applied to pick the nearest 5 to identify which activity it is.

1.4 Project Scope

The developed system worked on videos and evaluated on the database considered in our study. Four activities as discussed earlier are recognized by the system. Each video is assumed to contain a single individual only. The system may later be extended to more complex activities.

1.5 Organiztion of thesis

- Chapter No 2 Related to Literature Review belongs to our Proposed System.
- Chapter No 3 Related to Requirement Specification, in which we discuss about the technical and non-technical requirements.
- Chapter No 4 Related to Architecture Design, in which we discuss the designing of our System.
- Chapter No 5 Related to System Implementation, in which we discuss about the techniques and tools used in our project.
- Chapter No 6 Related to System testing and training, in which we discuss the test cases of our system as well as system outcomes or accuracy according to expected results.
- Chapter No 7 Related to Final conclusion and future work, which may be done in future, according to need.
- Chapter No 8 carries Research Paper References, which we are able to understand the actual problem and their solutions.

Chapter 2

Literature Review

2.1 Identify human activities on high-end phones using a word-friendly vector support

In [5],Activity-based computing aims to capture the state and environment of a user by using heterogeneous sensors to account for external computing resources. When these sensors are connected to the body, they allow continuous monitoring of many physiological signals. For example, this is an attractive use in medical applications. Use Peripheral Intelligence (AMI) to monitor the daily activities of the elderly. In this article, they present a system that uses inertial sensors in smartphones to identify human activity. Due to the limited power and computing power of these handsets, they offer a new, multi-category, hardware-based and user-friendly classification approach. This method adapts to the standard vector support machine (SFM) and uses a fixed-point calculation to reduce the cost of calculation. Compared to traditional MFS, computing costs are significantly increased while maintaining a similar accuracy, which helps to develop more sustainable systems for mothers.

2.2 Familiar with activities that accelerate user data annotation

In [6], they propose a new linear time method characteristics of the extraction of accelerometer signals in the system identifing Human Activities they use one to measure this method an acceleration based on an activity detection database is mentioned short tail - NAA. The results show that the method described training and test data come from a better performance in this case, the same person supports the linear support vector of the kernel. (SFM) and Radial Basic Function (RFP) based on comparable precision. Finally, they are able for real-time demonstration of the proposed method identify the activity on your phone with a three-axis feature accelerate. This method can be used to extract functions identify real activities on resources with limited resources.

2.3 Identify the activity of accelerometer data on the phone

In [7],Human movements can easily be displayed in real time, as a convenient tool for many applications and future applications. This article presents a real-time classification system for a basic human movement using a conventional handpiece equipped with an accelerometer. The purpose of this study is to verify the current real-time capabilities of traditional handsets and to implement all the algorithms necessary for pattern recognition to classify the associated human movement. Since the data processing server is not linked to this method, human monitoring is completely decentralized and requires only additional software to monitor people remotely. The feasibility of this approach opens up a new reasonable cost opportunity for the development of new applications.

2.4 Decision rules based on recognition of human activity

In [8],Recognition of human activities in the real world, there are many applications, such as observation, help robotics and simulation system. It includes many systems for recognizing human activities analyze static characteristics and location-based coordinates of activity detection. There are extras time series of information and frames in existing animation properties can be take action. This article uses dynamic and temporal data for comparison decision rules and activity recognition template. The human form is extracted using engineering model that covers several frames. The extracted form is converted to a binary state use aegean spatial mapping and canonical space conversion. Image data use the activity template to sample the images in a single filter window. This candidate the framework is compared to a model based model to cover the categories of activities Put the right word. 64 percent recognition of decision base model and activity models precision shows that this method can identify human activities.

2.5 Uniform framework for monitoring and analyzing human body movement

In [11], they propose a structure for dissecting non-inflexible movement, following and recognition in view of a movement design. These highlights are spoken to by HMM (Hidden Morkev Model) and spatial data. The low-level component extraction is combined with the worldly investigation, following issues and so forth. As per analysts, this is a high approach for some classes of human movement designs. Visual following is achieved by extricating the most successions of target areas from video stream utilizing a mix of irregular examining. These strategies enable us to perform critical errands, for example, human movement acknowledgment, walk examination, keyframe extraction and so on.

2.6 Motion Detection Based on Frame Difference Method

In [9], they are centered around building a framework that can watch people, and understanding their look, exercises and conduct by giving interface, and making sensible models for these reasons. This paper gives another calculation to recognizing moving articles from a still foundation in view of an edge contrast strategy. Right off the bat, the primary edge is taken and after some arrangement of casings second edge is removed. Also, the outright contrast is figured from the continuous casings and the outcome is put away in the framework. Thirdly, the outcome is changed over into dark scale and after that changed over into double. At last, the morphological channel is utilized to expel clamor from the last outcome.

2.7 Human Activity Recognition in Videos

In [10],As they need to arrange exercises on the premise of recordings so the issue emerges is there is such a significant number of things out of sight which make so this makes the execution of low level models and movement vectors unacceptable. So in this they have begun see and distinguishing the recordings of weight lifting, so they can without much of a stretch comprehend that if there must an entryway ringer in a video then this video is of weight lifting. However, in this the principle and vital errand is to distinguish the protest and its movement in video for better occasion grouping assignment. This require an exact discovery of question tracks in video. Be that as it may, this need a human aim to do as such. They address this by taking hopeful tracks from video and after that displaying them to make a model. In past work the tracks are extricated and a sack of highlights is separated from tracks. They likewise expressly removed human tracks from recordings which takes excessively time and a great deal of human consideration.

Model Formulation: In this they have to recognize activities from videos. So this recognition is helped by spatio-temporal position between object tubes provide good description of actions at high level. Extraction of good and required tube from video is main challenging task specially when no annotation is available on videos.

For experiments In this, they have to extract only human throughout the video. So they have used four techniques and compare their result and they choose best among them.

- Bag of words In this hog and hof features are extracted and vector quantization is done on SVM.
- Nibles at el. In this temporal alignment of different low level action is extracted to classify a class.
- Tang el at. In this HMM model is used to classify the video by semantic temporal segments in an event.
- No Latent In this they used features from high scoring tube to train SVM.

Chapter 3

Requirement Specifications

3.1 Functional requirements

3.1.1 Processing videos

Our application should cater almost all type of formats but just only recorded videos. As we have benchmark videos of our four activities, which we must recognize.

3.1.2 Input Video

Our application takes above mentioned videos as input and process them and return the result to user on application via a text box or a pop up message. This application should take videos which can be in the form of a video or a stream of images in a picture box.

3.1.3 MHI

MHI stands for motion history image, as we give a video to our application and then application process it as a series of image and in each image (frame) we analyze the motion of human and by analyzing in bunch of frame we make history of humans motion and name it as MHI.

3.1.4 Feature Detector

In this module we first extract useful feature of human in each frame which we need to identify to specify this motion a specific name or class for classification. By this feature we can measure or detect the variation of image point from a frame to frame.

HOG

Feature descriptor is basically works like an algorithm which takes input image and gives the feature vector.

In this system HOG descriptor is applied which makes the orientation based histogram based on gradient values. Each pixel in block gives vote for the gradient value.

Number of gradient point values depends on the size of block and pixels per block, by changing these values, gradient values also change. Selection of small or large block size depends how much detailed information we want to extract from image.

3.1.5 Classifier

We have used KNN classifier to identify the query video by analyzing the labels of first 5 values which is given by system. For more simplicity we have arranged the values in ascending order so that KNN must analyze first 5 labels and identify the activity performed in video.

3.2 Non Function Requirements

3.2.1 Availability

Our application is available for anyone and at every time when it is installed on computer as it is a desktop application.

3.2.2 Reliable

Application should provide authenticated and exact data about recognition of video which user want to recognize. User can also see how much given video resembles among 4 activities.

3.2.3 Security

This application can be used for security purpose further when its activities area extended to suspicious one.

3.2.4 Usability

When user wants to use it, the application should be ready to use. As use selects a video and press the button to check which kind of activity human is doing in that video so application process that video and quickly give result to user.

3.2.5 User Friendly

The application should be user friendly so that any person can use the application without any training or teaching. User can easily understand GUI while using the application.

3.2.6 Use case

Use case basically used to illustrate the user interaction with application while performing the function requirement. And how user will application after put into use.

Main Use Case

Our application has only two points of interaction with user.

- When user is selecting a video which user wants to identify which activity is performed.
- When application has done all processing and giving result to user through a text box or a pop up message.

3.3 Use Case

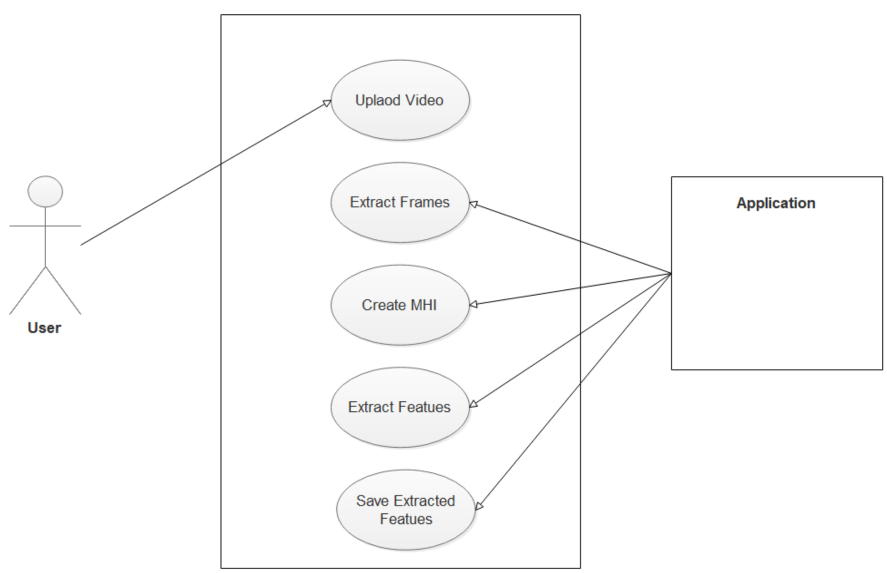

Figure 3.1: Training Use Case

In figure 3.1, use case describes the follow of our training sytem. first of all user upload a training dataset video, after that system extract frames and used them to create MHI, after that HOG Descriptor is used to extact the intrest points. Which are stored in bin to match the test videos.

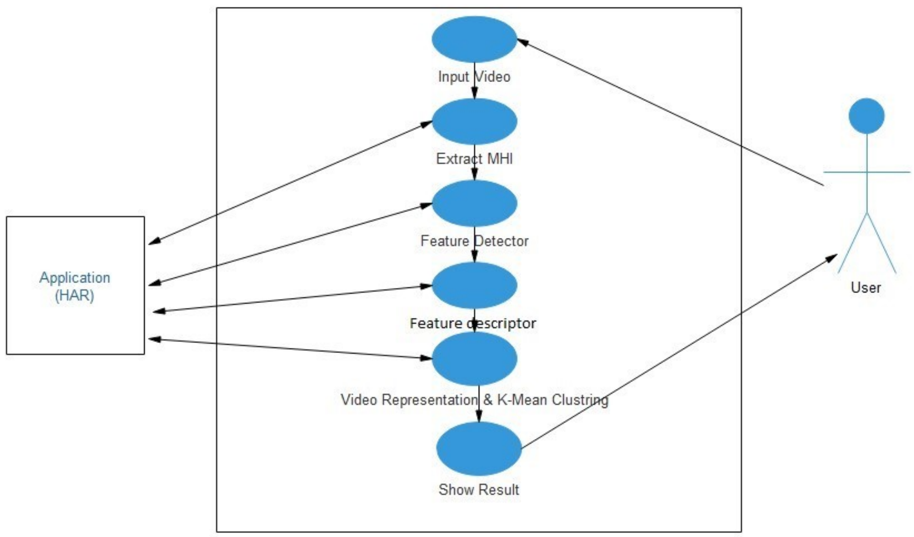

Figure 3.2: Testing Use Case

In figure 3.2, use case describes the follow of our sytem. first of all user upload a testing dataset video, after that system extract frames and used them to create MHI. After that HOG Descriptor is used to extact the intrest points. Which are used by KNN classifier to calculate the resultant activity.

3.3.1 Use Case 1: Start of Application

Figure 3.3: Start of Application

Table 3.1: Start of Application

Title	Start of Application
Description	User clicks on the application icon, then application must be a run-in time as soon as possible.
Primary Actor	User
Main Success Scenario	Application must be compatible with all MS Windows.
Pre Condition	Application must be installed in a system.
Post Condition	No Bug in running, and application cannot not get any type of exception.

3.3.2 Use Case 2: Uploading a Query Video

Figure 3.4: Upload a Query Video

Table 3.2: Upload a Query Video

Title	Upload a Query Video
Description	User clicks on the Upload Button to select a Query Video via Dialog Box.
Primary Actor	User
Main Success Scenario	Main Success Scenario Query Video must be run in an Application.
Pre Condition	Application must be in a running condition and have Data Sets of Query videos.
Post Condition	Further Processing will be done on the Query Video.

3.3.3 Use Case 3: Identify Activity

Figure 3.5: Identify Activity

Table 3.3: Identify Activity

Title	Identify Activity
Description	User clicks on the Identify Query Video Button to get the results from the system.
Primary Actor	User
Main Success Scenario	Results should be Accurate as Expected.
Pre Condition	Pre-Condition System should be well trained.
Post Condition	Result should be accurate and ready for the next Query Video.

Chapter 4

DESIGN

This chapter is based on different design aspect of our system.

4.1 System Architecture

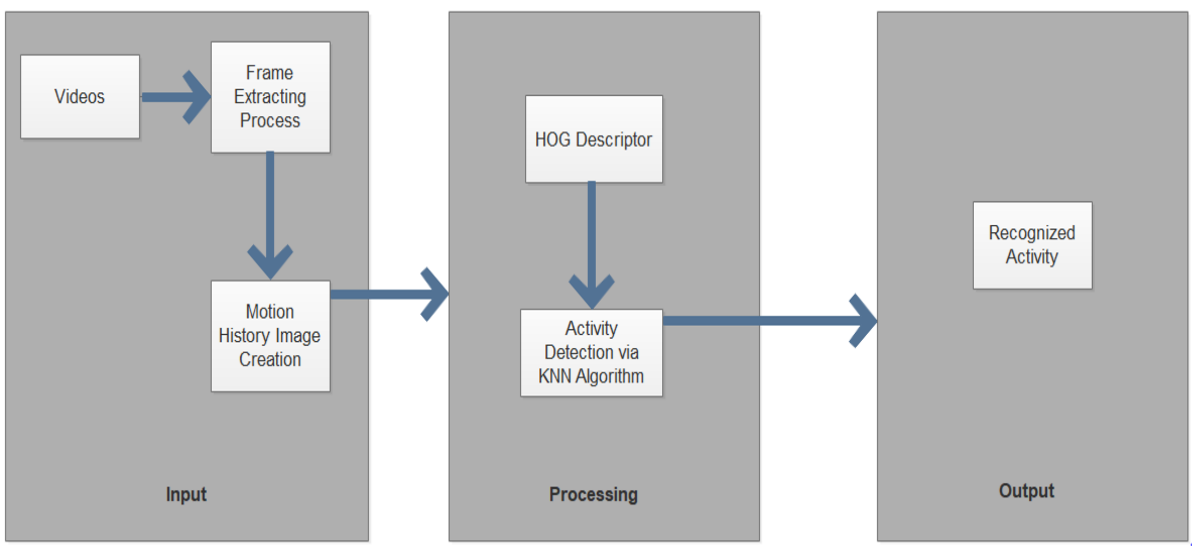

Figure 4.1: System Architecture

Our project is based on datasets (pre-recorded videos), hence no hardware requirements is needed for the project. Only EmguCV libraries are used for various purposes and KNN classification algorithm is used for machine learning purpose in the project.

4.2 Design Methodology

An overview of the intended system is presented. The aim of the proposed system is to analyze the video and detect the activity performed in the video from a set of pre-defined activities. The development of such a system targets the following activities.

1) Select video from the datasets.

2) Extract frames from the selected video.

3) Extract Motion History Image (MHI) based on the extracted frames.

4) Extract useful features to characterize different activities.

5) Train a model using these features via KNN classifier.

6) Test the model with testing dataset videos.

7) Get result from the system.

The decision will be based on a series of steps summarized in Figure 4.1. The first step, Extraction of frames, will be carried out through frame extraction Algorithm method, secondly, Motion History Image (MHI), will be carried out by taking absolute difference of extracted frames,an advantage of MHI is that although it is a representation of the history of pixel-level changes, only one previous frame needs to be stored.

$$H_t = \begin{cases} T & \text{if } D(x, y) = 1 \\ \max(0, H_t(x, y, t-1) - 1) & \text{otherwise} \end{cases}$$

After that useful features are extracted from MHI to detect different activities.

4.3 High Level Design

This section describes in further via designs and diagrams, how our project is worked.

4.3.1 Component Diagram

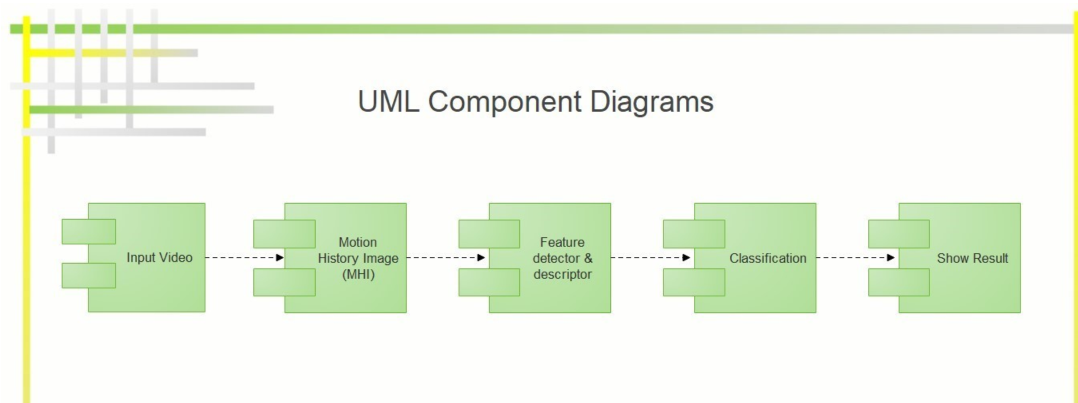

Figure 4.2: component Diagram

In figure 4.2, it describes the components used to perform these functions.

4.3.2 Activity Diagram

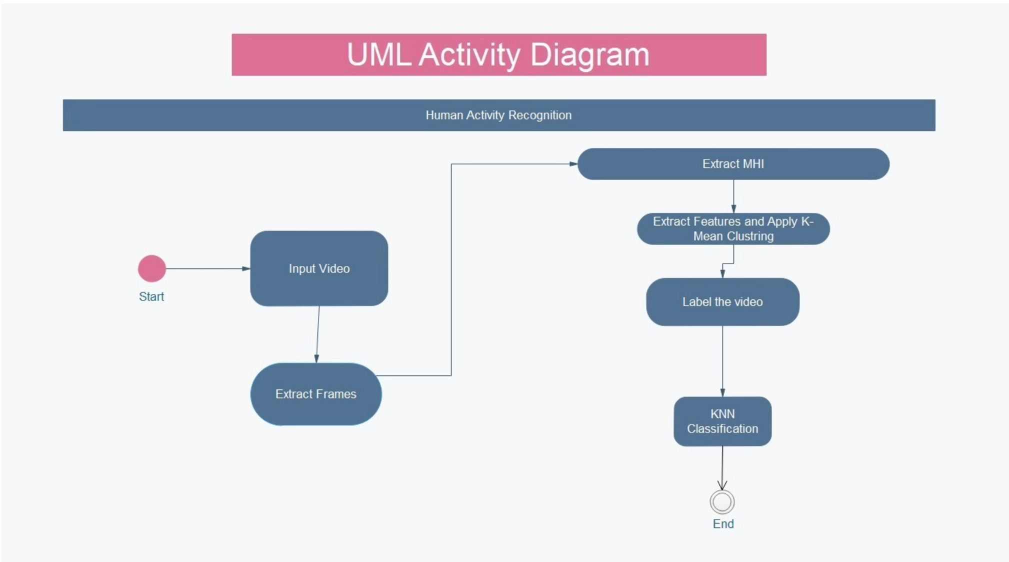

Figure 4.3: Activity Diagram

In figure 4.3, it describes the step wise activities performed in our system.

4.4 GUI Design

The Graphical User Interface should be simple and user friendly, so that everyone can use. The designed system will be useful not only for computer operators, but it can be used by any person that have some basic knowledge about computer, therefore the GUI had to be simple and easy to use. The developed interface shows the running video and after some interval of time it may show the result by processing on the input video. The GUI Design of our system is not final, so that it may possible to change the GUI design according to the need of the system.

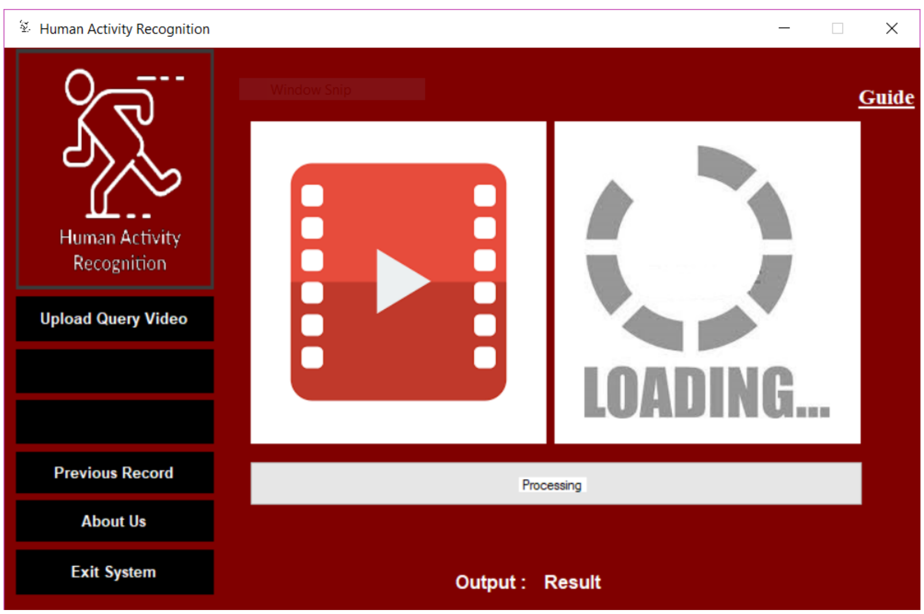

Figure 4.4: Starting Step

In figure 4.4, the GUI form is shown which will be seen when application starts. The logo is placed at top left most corner.

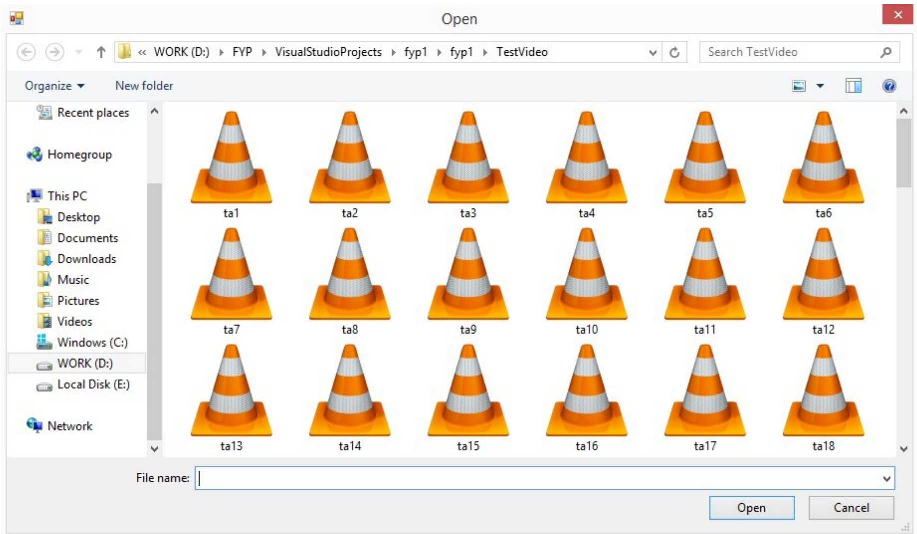

Figure 4.5: Show Dialog Box

In figure 4.5, the dialog box is open which provide the user to put any test video for query to identify which activity is performed in them.

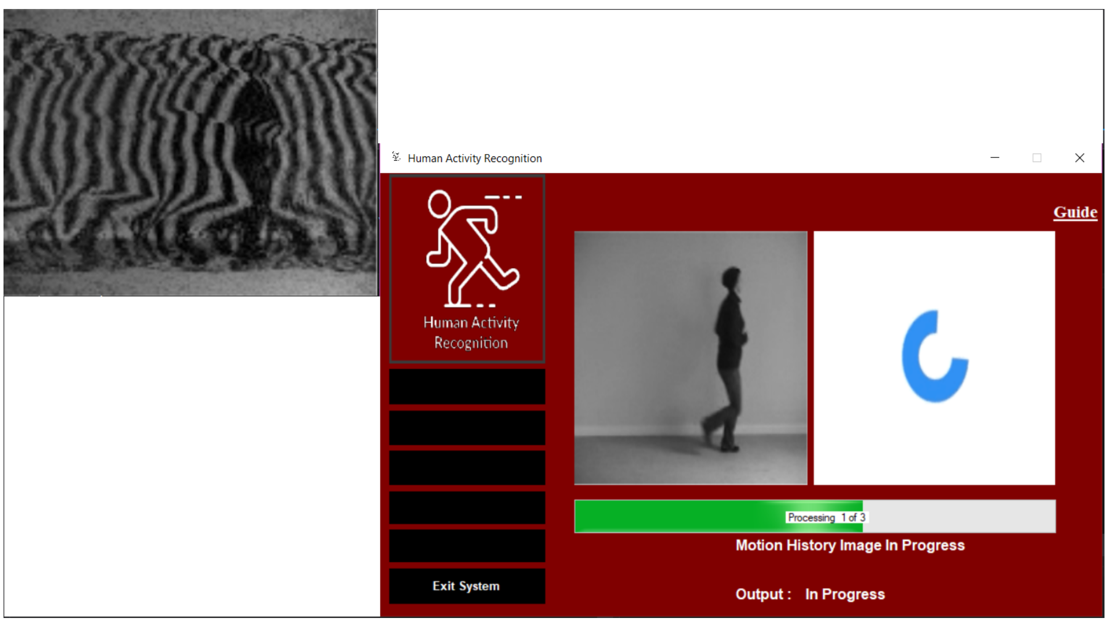

Figure 4.6: Running video on form

In figure 4.6, the application is running and making of MHI is also is progress in another small video and actual video is perform on main GUI form. Progress bar show the progress of MHI.

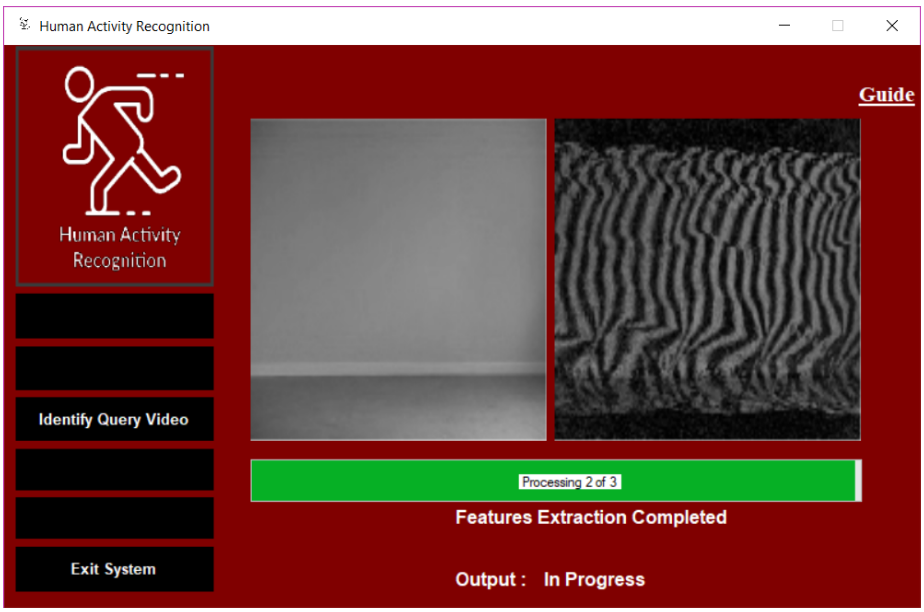

Figure 4.7: Feature Extraction

In figure 4.7, the progress bar 2/3 shows that HOG descriptor is applied on MHI extracted form test video and its values are extracted.

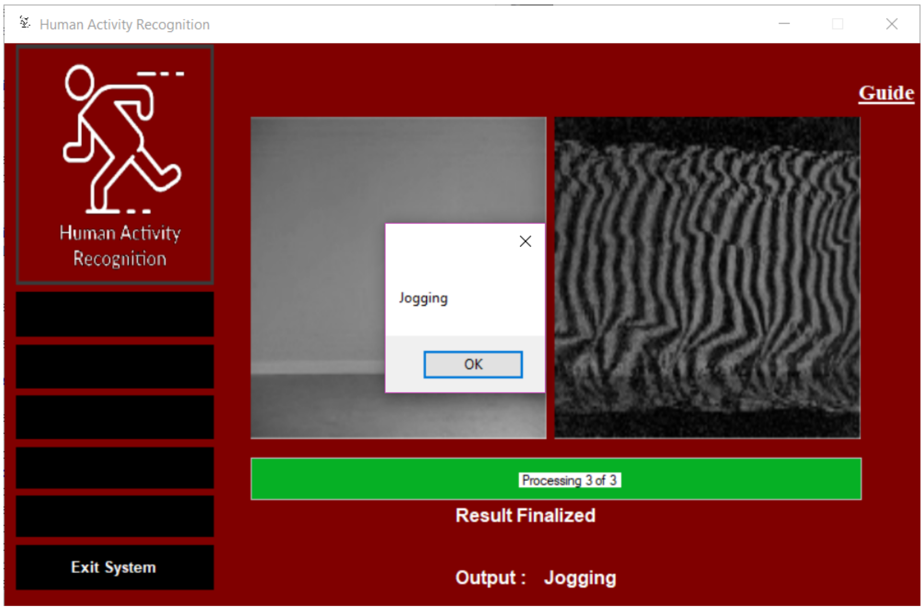

Figure 4.8: Final Result

In figure 4.8, results are finalized by application and identified activity in shown as message box as well as output label whereas progress bar 3/3 show that application has finished it processing on given video.

Chapter 5

System Implementation

Implementation is the process of transforming ideas from concept to reality. The system is implemented by programming and publishing, implementing technical specifications or algorithms as programs or software components or other computer systems.

5.1 Introduction

In system Architecture, three major units are considered i.e. input, output and processing. In input unit, user upload a query video in a system while a processing unit is consisting of pre-processing in which motion history image (MHI) is created and give an image of MHI. After feature extraction and machine learning, result will be shown in output unit. These results are stored in database for further usage or record. System also guide a new user, by viewing a sample video of running system, so that every person can use this system for various purposes.

5.1.1 Algorithmic Workflow

The overall work flow of the system is defined in the diagram presented in figure 5.1.

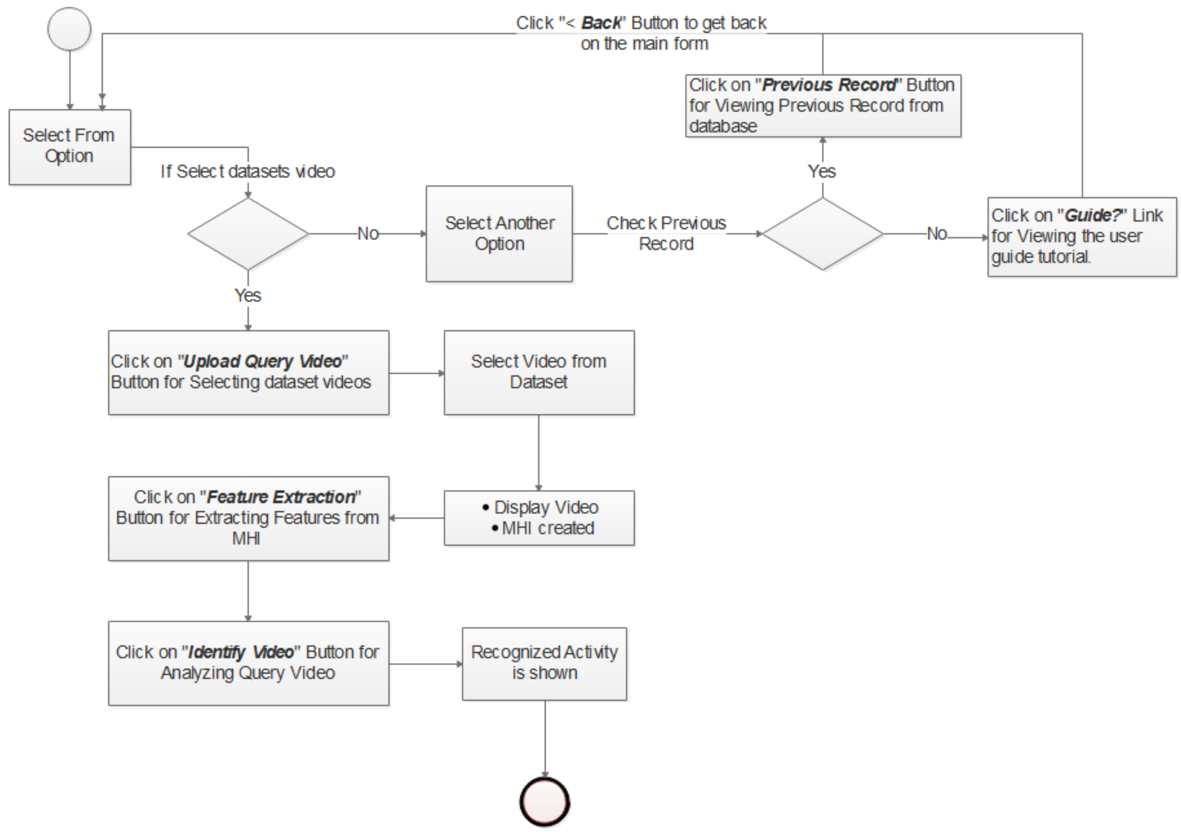

Figure 5.1: Algorithm Work Flow

5.2 System Components

The following modules or components from the system are viewed.

- Select a Query video.
- Taking difference between two frames (applied on whole video), MHI is created.
- Apply HOG Descriptor on MHI.
- Add labels.
- Apply KNN classification, results are to be shown based on learning.

5.2.1 Selection of video

In this phase, user upload a Query video in a system by clicking on a button Upload Query Video, after that dialog box will open and user select a desired video to be recognized.

5.2.2 Frame Difference

In this phase, system reads a video uploaded by user, and compute a difference between two frames by current frame from previous frame, which is applied on all the frames of the video, the system can show the results of running video and differences on the running form.

$$\|Fi - F(i-1)\|$$

Where F denotes a video of a frame, i denotes the index of the frame. Frame differencing produce a history image of motion done in respected video. In our implementation, instead of taking the difference of after every 5 or 10 frames, we take the difference of every frame so that MHI will be created smoothly and information cannot be loss. And, the feature detection is easier to accessible.

5.2.3 Feature Extraction

There are different ways to detect human activity. One of them is HOG, which is mainly used for human detection. In our study, we found that the results of the HOG signature are very fast and accurate compared to other methods.

After Motion History Image (MHI) will be created, HOG descriptor is applied on the image to extract points. In human detection, for detection of human the main things that are important are extracting the most relevant information from obtainable images and human classification. HOG provides

- Good presentation of human contour.
- HOG is steady to lightening changes and small image movements.
- HOG can be computed in a continual period.

Method is same as edge oriented histogram, shape context etc. The characteristics of the pig are extracted from the gray image. These characteristics are returned in the vector 1-By-N, where N is the length of the HOG function. The returned function is used to encode local shape information in the image. This information can be used for monitoring, detection and classification purposes.
Computation of Gradient First step of calculation in feature detection is to normalize the color and gamma points.However these feature can be extracted via HOG descriptor computation. In our system our data sets are already In gray scale and have fixed width and height of frames.so that our system has no need for per processing. The most common method is the 1-D centered, points are in both horizontal and vertical directions.

Orientation Binning This is the second step, in our system i.e. Human activity Recognition is carried out using the well-known histogram of oriented gradients (HOG) method. The image region is dividedinto cells and for each cell gradient magnitude and direction is computed. The orientationsof gradients within a cell are counted in a 9-bin histogram (0 to 180 degrees) and thehistograms of all cells are concatenated together to generate the HOG descriptor. In our system we use 32x32 pixels per block and 2x2 cells per block with 9 histogram channels. Thedescriptor is then used to distinguish different activities performed in videos which are used in our system.

In tests the gradient magnitude automatically produces the best result. Other options could include the square root or square of the gradient magnitude or some clip version.

5.2.4 Activity classification

In our system, a KNN classifier is learned with HOG feature vector. That any N training vector is given and KNN algorithm identifies the k-nearest neighbours of class c, regardless of labels. The learning example is a multidimensional feature, each with a class tag. This phase of the algorithm depends only on the storage characteristics and tags of the class. In our system we use KTH datasets for training and testing with a ratio of 80:20. We have 100 videos per activity, hence in a single class our system has 80 videos for training and 20 videos for testing purpose. A commonly used distance measuring algorithm is Euclidian distance, so we use this distance measuring algorithm to calculate the distances. After getting distance of each testing video from training data we apply k=5, which gives us the nearest 5 neighbours of trained data by which we can easily identify that which activity is performed in test video. So, the result calculated by KNN is displayed as output on display screen.

5.3 Tools and Technologies

The system is developed in c# using Microsoft visual studio professional. For processing of video frames, extraction of features i.e. Histogram of Oriented Gradients (HOG) as so on, we use EmguCV libraries.

EmguCV is an open source image processing library which in-fact is a wrapper of classical OpenCV library used in c++.

After the detailed description of our system implementation and algorithm, we present the results of our systematical experimental evaluation in next chapter i.e. Chapter No 6.

Chapter 6

System Testing and Evaluation

For testing of our system, as we worked on benchmark datasets (KTH), we have taken 20% of videos for testing of our system. Which means that we have taken randomly 20 videos from each activity for testing which are not present in a training of a system. The following section present the results of our system i.e. “human activity recognition”.

6.1 Motion History Image Creation

For testing we first take a video from a folder where all testing videos are placed separately from training data. Then we make MHI from a video to extract the motion history of testing video so that we can extract the activity history to further process and identify the activity. Each MHI takes approximately same time as time taken by a video to play. Approximation time of each video = 20 30 seconds.

6.2 HOG Creation

After the MHI creation, HOG would be done and the gradients points made by the HOG are stored in a file. So that application could identify the activity on the basis of learning which was done on training data videos.

Table 6.1, 6.2, 6.3 and 6.4 tells us about the actual and expected outcomes of clapping, boxing, waving and jogging activities respectively.

Table 6.1: Clapping Outcome

Sr.No.	**Test Video No.**	**Actual Result**	**Remarks**
1	Ta1	Clapping	Pass
2	Ta2	Clapping	Pass
3	Ta3	Clapping	Pass
4	Ta4	Clapping	Pass
5	Ta5	Clapping	Pass
6	Ta6	Clapping	Pass
7	Ta7	Clapping	Pass
8	Ta8	Clapping	Pass
9	Ta9	Clapping	Pass
10	Ta10	Clapping	Pass
11	Ta11	Clapping	Pass
12	Ta12	Clapping	Pass
13	Ta13	Clapping	Pass
14	Ta14	Clapping	Pass
15	Ta15	Clapping	Pass
16	Ta16	Clapping	Pass
17	Ta17	Clapping	Pass
18	Ta18	Clapping	Pass
19	Ta19	Boxing	Fail
20	Ta20	Clapping	Pass
Total: 20	Ta1-Ta20	19	95%

This table shows the overall results of clapping activity, in which 19 out of 20 videos are recognized in a true manner. The percentage of our system for clapping activity is 95%.

Table 6.2: Boxing Outcomes

Sr.No.	**Test Video No.**	**Actual Result**	**Remarks**
1	Tb1	Boxing	Pass
2	Tb2	Boxing	Pass
3	Tb3	Boxing	Pass
4	Tb4	Boxing	Pass
5	Tb5	Boxing	Pass
6	Tb6	Boxing	Pass
7	Tb7	Boxing	Pass
8	Tb8	No Result	Fail
9	Tb9	Boxing	Pass
10	Tb10	Boxing	Pass
11	Tb11	Boxing	Pass
12	Tb12	Boxing	Pass
13	Tb13	Boxing	Pass
14	Tb14	Boxing	Pass
15	Tb15	Boxing	Pass
16	Tb16	Boxing	Pass
17	Tb17	Clapping	Fail
18	Tb18	Boxing	Pass
19	Tb19	Boxing	Pass
20	Tb20	Boxing	Pass
Total : 20	Tb1-Tb20	18	90%

This table shows the overall results of boxing activity, in which 18 out of 20 videos are recognized in a true manner. The percentage of our system for boxing activity is 90%.

Table 6.3: Waving Outcomes

Sr.No.	**Test Video No.**	**Actual Result**	**Remarks**
1	Tc1	Waving	Pass
2	Tc2	Waving	Pass
3	Tc3	Waving	Pass
4	Tc4	Boxing	Fail
5	Tc5	Waving	Pass
6	Tc6	Waving	Pass
7	Tc7	Waving	Pass
8	Tc8	Waving	Pass
9	Tc9	Waving	Pass
10	Tc10	Waving	Pass
11	Tc11	Waving	Pass
12	Tc12	Waving	Pass
13	Tc13	Waving	Pass
14	Tc14	Waving	Pass
15	Tc15	Clapping	Fail
16	T16	Boxing	Fail
17	Tc17	Clapping	Fail
18	Tc18	Waving	Pass
19	Tc19	Waving	Pass
20	Tc20	Waving	Pass
Total: 20	Tc1-Tc20	16	80%

This table shows the overall results of waving activity, in which 16 out of 20 videos are recognized in a true manner. The percentage of our system for boxing activity is 80%.

Table 6.4: Jogging Outcomes

Sr.No.	**Test Video No.**	**Actual Result**	**Remarks**
1	Td1	Jogging	Pass
2	Td2	Jogging	Pass
3	Td3	Jogging	Pass
4	Td4	Jogging	Pass
5	Td5	Jogging	Pass
6	Td6	Jogging	Pass
7	Td7	Jogging	Pass
8	Td8	Jogging	Pass
9	Td9	Jogging	Pass
10	Td10	Jogging	Pass
11	Td11	Jogging	Pass
12	Td12	Jogging	Pass
13	Td13	Jogging	Pass
14	Td14	Jogging	Pass
15	Td15	Jogging	Pass
16	Td16	Jogging	Pass
17	Td17	Jogging	Pass
18	Td18	Jogging	Pass
19	Td19	Jogging	Pass
20	Td20	Boxing	Pass
Total : 20	Td1 – Td20	20	100%

This table shows the overall results of boxing activity, in which 20 out of 20 videos are recognized in a true manner. The percentage of our system for boxing activity is 100%.

6.2.1 Results Summary

Table 6.5: Overall Results

Activity	**Clapping**	**Boxing**	**Waving**	**Jogging**	**No Results**
Clapping	**19**	1	0	0	0
Boxing	1	**18**	0	0	1
Waving	2	2	**16**	0	0
Jogging	0	0	0	**20**	0

Over all accuracy of our system based on the results shown in table 6.5 is 91.25%.

6.3 Usability Testing

In usability testing we have to ensure that what steps must need to be followed so that user can easily use the system. This system is safe and easy to use and easy to learn for new user due to its simplicity. In our system we have also provided a guide video for easy and quick learning for our new users.

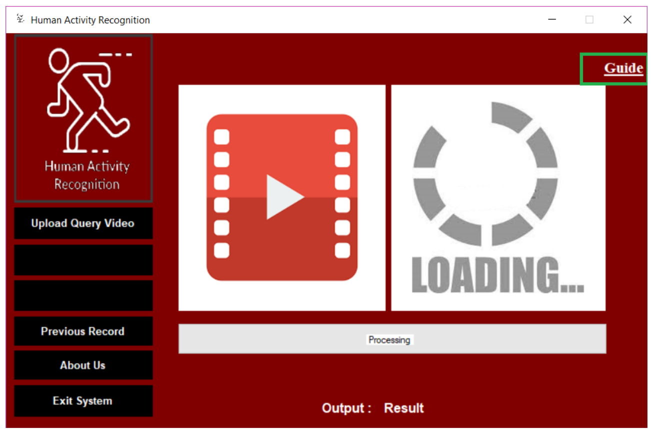

Figure 6.1: Guide Link

In figure 6.1, it shows the hyperlink design in our system for video tutorial for our new users.

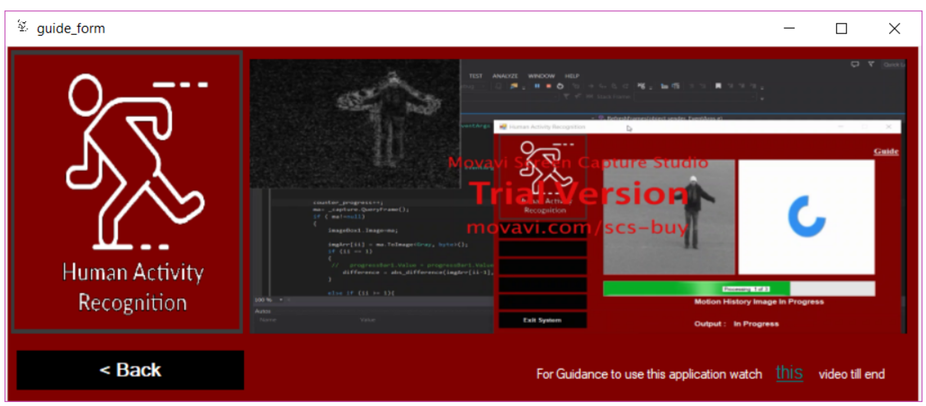

Figure 6.2: Guide Form

In figure 6.2, form shows the guide form design in our system for playing a tutorial video.

6.4 Compatibility Testing

In compatibility testing we make sure that our application work on different platform with which it is compatible and with which it is not. As this application is on Microsoft visual studio using C# and EmguCV libraries and as a windows form application so it is compatible with all version of windows.

This application is developed using Visual Studio 2013 professional and EmguCV version 3.1.

6.5 Software Performance Testing

Software Performance testing is done to make sure that either this application is efficient and reliable or not. This application efficiently do its all tasks which are loading frame to play video, making MHI form videos, making HOG gradients and identify the activity in query video and displaying the outcome on screen.

6.6 Graphical User Interface Testing

The most important module of any application is its Graphical User Interface because it is very important to satisfy and attain the user attention in application. In our application we have tested the GUI for better user understanding and ease of use for user. Interface testing done by some test cases.

6.7 Test Cases

Test cases are some conditions and constraints under which it is make sure that our system perform correctly and efficiently.

6.7.1 Application startup test case

This test case (table 6.1) shows that the application start-up correctly on different windows.

Table 6.6: Application startup test case

Test case id	01	
Description	**Startup test**	
Compatibility	Should run on all MS Windows	
Requirements	Requirement No. 01	
Initial Condition	None	
Step	**Task and expected result**	**Remarks**
1	Open the application	Pass
2	Make sure application runs on multiple systems	Pass
3	Upload query videos	Pass
4	Make sure all buttons working correctly	Pass

6.7.2 Query video loading test case

This test case (table 6.2) shows that the system successfully load a query video.

Table 6.7: Query video loading test case

Test case id	02	
Description	**Test the functionality of loading video**	
Compatibility	Should run on any system	
Requirements	Requirement No. 02	
Initial Condition	Test datasets videos available in system	
Step	**Task and expected result**	**Remarks**
1	Open the form to upload a query video	Pass
2	Ensure that dialog box open when click on upload button.	Pass
3	Datasets are shown in dialog box and able to select a video	Pass
4	Video is running on form.	Pass

6.7.3 Identify query video test case

This test case (table 6.3) shows that the system successfully identifies the activity performed in a query video.

Table 6.8: Identify query video test case

Test case id	03	
Description	**Test the system functionality to identify activity**	
Compatibility	Should run on any system	
Requirements	Requirement No. 03	
Initial Condition	Test datasets videos available in system	
Step	**Task and expected result**	**Remarks**
1	Video is running on a form.	Pass
2	Creation of MHI.	Pass
3	Creation of HOG for gradient points and stored in a text file.	Pass
4	Result shown in a label as well as message box	Pass

Chapter 7

Conclusion & Future Work

7.1 Conclusion

Security and surveillance have gained signicant importance during the last decade. There has been an increasing trend towards the installation of surveillance cameras at public places including airports, bus stations, shopping centers and important roads. So that we develop a system, that can recognize different pre-defined activities i.e. Boxing, Waving, Clapping, Jogging. Given a video, our system can create a MHI (Motion History Image) frame by frame, then features will be extracted via HOG implementation that give us a 1D float array that carries a gradient point. After that machine learning via KNN classification will be applied.

We use KTH datasets for the implementation of our proposed system. It carries 100 videos per activity in total 400 videos are used for training and testing of our system, we use 80:20 ratio for training and testing of our system.

7.2 Future Work

Future work on this system may consider more activities and real-time recording on the system. Other queries like multiple activities performed on a single video or density-weighted methods may be used to increase the efficiency of our system and more reliable to use. On present day we use offline videos, it is more interesting to recognize from online videos. Hence storage capacity will also be increased and so on.

Bibliography

[1] Kim, Eunju, SumiHelal, and Diane Cook. "Human activity recognition and pattern discovery." IEEE Pervasive Computing 9.1 (2010).

[2] Sadanand, Sreemanananth, and Jason J. Corso. "Action bank: A high-level representation of activity in video." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[3] Turaga, Pavan, et al. "Machine recognition of human activities: A survey." IEEE Transactions on Circuits and Systems for Video Technology 18.11 (2008): 1473-1488.

[4] http://www.nada.kth.se/cvap/actions/

[5] http://www.la-press.com/redirect_file.php?fileId=1180&filename=BMI-3-Foster-et-al&fileType=pdf

[6] Huang P S, Harris C J, and Nixon M S, Human gait recognition in canonical space using temporal templates, IEEE Proc. VISP 142 1999 93100.

[7] Krahnstover N, Yeasin M, and Sharma R, Towards a unified framework for tracking and analysis of human motion, in Proc. of IEEE Workshop on Detection and Recognition of Events in Video (2001) 47-54.

[8] Yamato J, Ohya J, and Ishii K, Recognizing human action in time sequential images using Hidden Markov Model, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (1992) 379-385.

[9] Davis J W and BobickA F, The representation and recognition of human movement using temporal templates, in Proc. of IEEE Computer Vision and Pattern Recognition (1997) 928-934

[10] Polana R and Nelson R, Detection and recognition of periodic, nonrigid motion, International Journal of Computer Vision 23(3) (1997) 261-282.

[11] Krahnstover N, Yeasin M, and Sharma R, Towards a unified framework for tracking and analysis of human motion, in Proc. of IEEE Workshop on Detection and Recognition of Events in Video (2001) 47-54.

[12] Singla, Nishu. "Motion detection based on frame difference method." International Journal of Information & Computation Technology 4.15 (2014): 1559-1565.

[13] Goel, VigneshRamanathanAnkurSarinRishabh. "Human Activity Recognition in Videos."