Chapter-1

INTRODUCTION

1.1 Introduction

Recognizing human activities for video surveillance is one of the most promising applications of computer vision. In recent years, this problem has caught the attention of researchers from industry, academia, security agencies, consumer agencies and the general populace too. Video surveillance is of increasing importance to many applications, such as elder-care, home-nursing, and unusual event alarming [1].

The task of recognizing human action poses several challenges. Human action is extremely diverse, and to build a system that can be used to successfully identify any type of action is a serious problem indeed. An interesting fact about human activity is the inherent similarity in the way actions are carried out. That is, people jump, stand, walk, bend down and get up in a more or less similar fashion, assuming, of course, there is no impediment in the performance of these actions.

Most systems that perform human motion analysis address general common tasks, such as: person detection & tracking, activity classification, behaviour interpretation and also person identification. Obviously, although some of these tasks can be considered independently, they must be solved in a common framework, where information can be communicated and exchanged between the different system modules. As the detection and tracking systems have progressed significantly in the past few years, [17, 22, 19], human motion and behaviour interpretation have naturally become the following step. In a surveillance scenario, tracking is the very first step and behaviour recognition the final goal. The task of activity recognition can be viewed as a bridge between the pixel measurements, given by the tracker, and a more abstract behaviour description. In this paper, we focus on this intermediate level, that is essential to achieve the desired final large-scale interpretation. The need for such systems is increasing everyday with the number of surveillance cameras deployed in public spaces. Needless to say, the “traditional” job of the security operator, monitoring several video streams for extended periods of time, becomes impossible, as the

number of cameras grows exponentially. Instead, we need systems able to detect, categorize and recognize human activity, calling for human attention only when necessary.

There are so many methods have been developed to recognize the activities and many other methods are in the process. So for this cause we are also introducing a new method for recognizing the human activities. We are basically applying our method for activity recognitions. We have applied this method to recognize the different activities for the video surveillance. We have taken the three types activity performed by the person walking, standing and jumping. The rest of the paper is structured as follows: Section 2 highlights the literature survey. Section 3 describes the research aims and objectives. Methodology for activity recognition for video surveillance is described in Section 4. Section 5 and 6 describe the classifiers for the activity recognition system and experimental results and conclusion respectively.

1.2 Background

Before discussing activity recognition, it is needed to know the basic terms used in this project. This work comes under pattern recognition and machine vision. So a brief overview of these terms has been given in the next section.

1.2.1 Pattern Recognition

Automatic (machine) recognition, description, classification, and grouping of patterns are important problems in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, artificial intelligence, and remote sensing. But what is a pattern? A pattern is defined as opposite of a chaos; it is an entity, vaguely defined, that could be given a name. For example, a pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal. Given a pattern, its recognition/classification may consist of one of the following two tasks: 1) supervised classification (e.g. discriminant analysis) in which the input pattern is identified as a member of a predefined class, 2) unsupervised classification (e.g., clustering) in which the pattern is assigned to a hitherto unknown class. Note that the recognition problem here is being posed as a classification or categorization task, where the classes are either defined by the system

designer (in supervised classification) or are learned based on the similarity of patterns (in unsupervised classification). Interest in the area of pattern recognition has been renewed recently due to emerging applications which are not only challenging but also computationally more demanding.

The rapidly growing and available computing power, while enabling faster processing of huge data sets, has also facilitated the use of elaborate and diverse methods for data analysis and classification. At the same time, demands on automatic pattern recognition systems are rising enormously due to the availability of large databases and stringent performance requirements (speed, accuracy, and cost). In many of the emerging applications, it is clear that no single approach for classification is optimal and that multiple methods and approaches have to be used. Consequently, combining several sensing modalities and classifiers is now a commonly used practice in pattern recognition.

The design of a pattern recognition system essentially involves the following three aspects: making. The problem domain dictates the choice of sensor(s), pre-processing technique, representation scheme, and the decision making model. It is generally agreed that a well-defined and sufficiently constrained recognition problem (small intra-class variations and large interclass variations) will lead to a compact pattern representation and a simple decision making strategy. Learning from a set of examples (training set) is an important and desired attribute of most pattern recognition systems. The four best known approaches for pattern recognition are: 1) template matching, 2) statistical classification, 3) syntactic or structural matching, and 4) neural networks. These models are not necessarily independent and sometimes the same pattern recognition method exists with different interpretations.

1.2.2 Machine learning

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviours given all possible inputs is too large to be covered by

the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases.

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield absolute guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. In addition to performance bounds, computational learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time. There are many similarities between machine learning theory and statistics, although they use different terms.

Algorithms

Machine learning algorithms can be organized into a taxonomy based on the desired outcome of the algorithm.

A. Supervised learning

B. Unsupervised learning

C. Semi-supervised learning

D. Reinforcement learning

A. Supervised learning

Let us begin by considering the simplest machine learning task: supervised learning for classification.

Suppose we wish to develop a computer program that, when given a picture of a person, can determine whether the person is male or female. Such a program is called a classifier, because it assigns a class (i.e., male or female) to an object (i.e., a photograph). The task of supervised learning is to construct a classifier given a set of classified training examples—in this case, example photographs along with the correct classes. The key challenge for supervised learning is the problem of generalization: After analyzing only a (usually small) sample of photographs, the learning system should output a classifier that works well on all possible photographs. A pair consisting of an object and its associated class is called a labelled example. The set of labelled examples provided to the learning algorithm is called

the training set. Suppose we provide a training set to a learning algorithm and it outputs a classifier. How can we evaluate the quality of this classifier? The usual approach is to employ a second set of labelled examples called the test set. We measure the percentage of test examples correctly classified (called the classification rate) or the percentage of test examples misclassified (the misclassification rate). The reason we employ a separate test set is that most learned classifiers will be very accurate on the training examples. Indeed, a classifier that simply memorized the training examples would be able to classify them perfectly. We want to test the ability of the learned classifier to generalize to new data points. Note that this approach of measuring the classification rate assumes that each classification decision is independent and that each classification decision is equally important. These assumptions are often violated. The independence assumption could be violated if there is some temporal dependence in the data. Suppose for example, that the photographs were taken of students in classrooms. Some classes (e.g., early childhood development) primarily contain girls, other classes (e.g., car repair) primarily contain boys. If a classifier knew that the data consisted of batches, it could achieve higher accuracy by trying to identify the point at which one batch ends and another begins. Then within each batch of photographs, it could classify all of the objects into a single class (e.g., based on a majority vote of its guesses on the individual photographs). These kinds of temporal dependencies arise frequently. For example, a doctor seeing patients in a clinic knows that contagious illnesses tend to come in waves. Hence, after seeing several consecutive patients with the flu, the doctor is more likely to classify the next patient as having the flu too, even if that patient's symptoms are not as clear cut as the symptoms of the previous patients. The assumption of equal importance could be violated if there are different costs or risks associated with different misclassification errors. Suppose the classifier must decide whether a patient has cancer based on some laboratory measurements. There are two kinds of errors. A false positive error occurs when the classifier classifies a healthy patient as having cancer. A false negative error occurs when the classifier classifies a person with cancer as being healthy. Typically false negatives are more costly than false positives, so we might want the learning algorithm to prefer classifiers that make fewer false negative errors, even if they make more false positives as a result.

The term supervised learning includes not only learning classifiers but also learning functions that predict numerical values. For example, given a photograph of a person, we might want to predict the person's age, height, and weight. This task is usually called regression. In this case, each labelled training example is a pair of an object and the associated numerical value. The quality of a learned prediction function is usually measured as the square of the

difference between the predicted value and the true value, although sometimes the absolute value of this difference is measured instead.

B. Unsupervised learning

In machine learning, unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to pre-process data.

Approaches to unsupervised learning include:

• clustering (e.g., k-means, mixture models, k-nearest neighbours, hierarchical clustering),

• Blind signal separation using feature extraction techniques for dimensionality reduction (e.g., Principal component analysis, Independent component analysis, Non-negative matrix factorization, Singular value decomposition).

Among neural network models, the self-organizing map (SOM) and adaptive resonance theory (ART) are commonly used unsupervised learning algorithms. The SOM is a topographic organization in which nearby locations in the map represent inputs with similar properties. The ART model allows the number of clusters to vary with problem size and lets the user control the degree of similarity between members of the same clusters by means of a user-defined constant called the vigilance parameter. ART networks are also used for many pattern recognition tasks, such as automatic target recognition and seismic signal processing.

C. Semi-supervised learning

In computer science, semi-supervised learning is a class of machine learning techniques that make use of both labelled and unlabeled data for training - typically a small amount of labelled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data). Many machine-learning researchers have found that unlabelled data, when used in conjunction with a small amount of labelled data, can produce considerable improvement in learning accuracy. The acquisition of labelled data for a learning problem often requires a skilled human agent to manually classify training examples.

The cost associated with the labelling process thus may render a fully labelled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value.

One example of a semi-supervised learning technique is co-training, in which two or possibly more learners are each trained on a set of examples, but with each learner using a different, and ideally independent, set of features for each example. An alternative approach is to model the joint probability distribution of the features and the labels. For the unlabelled data the labels can then be treated as 'missing data'. Techniques that handle missing data, such as Gibbs sampling or the EM algorithm, can then be used to estimate the parameters of the model.

D. Reinforcement learning

Reinforcement learning is an approach to artificial intelligence that emphasizes learning by the individual from its interaction with its environment. This contrasts with classical approaches to artificial intelligence and machine learning, which have downplayed learning from interaction, focusing instead on learning from a knowledgeable teacher, or on reasoning from a complete model of the environment. Modern reinforcement learning research is highly interdisciplinary; it includes researchers specializing in operations research, genetic algorithms, neural networks, psychology, and control engineering.

Reinforcement learning is learning what to do-how to map situations to actions-so as to maximize a scalar reward signal. The learner is not told which action to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation, and through that all subsequent rewards.

These two characteristics-trial-and-error search and delayed reward-are the two most important distinguishing features of reinforcement learning. One of the challenges that arise in reinforcement learning and not in other kinds of learning is the trade off between exploration and exploitation. To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover which actions these are it has to select actions that it has not tried before. The agent has to exploit what it already knows in order to obtain reward, but it also has to explore in order to make better action selections in the future. The dilemma is that neither exploitation nor exploration can be pursued exclusively without failing at the task.

1.2.3 Computer vision

Activity recognition is a sub-part of computer vision. So it is needed to know the basics of computer vision. Computer vision (image understanding) is a discipline that studies how to reconstruct, interpret and understand a 3D scene from its 2D images in terms of the properties of the structures present in the scene. Computer vision is concerned with modelling and replicating human vision using computer software and hardware. It combines knowledge in computer science, electrical engineering, mathematics, physiology, biology, and cognitive science. It needs knowledge from all these fields in order to understand and simulate the operation of the human vision system.

1.2.3.1 Computer Vision Hierarchy

- Low-level vision: process image for feature extraction (edge, corner, or optical flow).
- Intermediate-level vision: object recognition and 3D scene interpretation using features obtained from the low-level vision.
- High-level vision: interpretation of the evolving information provided by the intermediate level vision as well as directing what intermediate and low level vision tasks should be performed. Interpretation may include conceptual description of a scene like activity, intention and behaviour.

1.2.3.2 Why study Computer Vision?

• Images and movies are everywhere
• Fast-growing collection of useful applications
– building representations of the 3D world from pictures
– Automated surveillance (who's doing what)
– Movie post-processing
– face recognition
• Various deep and attractive scientific mysteries
– How does object recognition work?

Chapter-2

LITERATURE SURVEY

Much work has been done in activity recognition. Cai and Aggarwal [2] discuss the different approaches used in the recognition of human activities. They classify the approaches towards human activity recognition into state-space and template matching techniques. Liao et al [3] discuss methodologies which use motion in the recognition of human activity. Ayers and Shah [4] have developed a system that makes context-based decisions about the actions of people in a room. These actions include entering a room, using a computer terminal, opening a cabinet, picking up the phone, etc. Their system is able to recognize actions based on prior knowledge about the layout of the room. Davis, Intille and Bobick [10] have developed an algorithm that uses contextual information to simultaneously track multiple, non-rigid objects when erratic movements and object collisions are common. However, both of these algorithms require prior knowledge of the precise location of certain objects in the environment. In [4], the system is limited to actions like sitting and standing. Also, it is only able to recognize a picking action by knowledge of where the object is and tracking it after the person has come within a certain distance of it. In [8], Davis uses temporal plates for matching and recognition. The system computes history images (MHI's) of the persons in the scene. Davis [8] computes MHI's for 18 different images in 7 different orientations. These motion images are accumulated in time and form motion energy images (MEI's). Moment-based features are extracted from MEI's and MHI's and employed for recognition using template matching. Although template matching procedures have a lower computational cost, they are usually more sensitive to the variance in the duration of the movement. A number of researchers have attempted the full three-dimensional reconstruction of the human form from image sequences, presuming that such information is necessary to understand the action taking place [11, 7, 15]. Others have proposed methods for recognizing action from the motion itself, as opposed to constructing a three-dimensional model of the person and then recognizing the action of the model [12, 5]. Rosario and Pentland [13], uses the Bayesian framework for modelling human actions. Given the correct probability density functions, Bayes theory is optimal in the sense of producing minimal classification errors. State space models have been widely used to detect, predict and estimate time series over a long period of time. Many state space systems use the hidden Markov model (HMM), a probabilistic

model for the study of discrete time series. In [13, 16], HMMs have been applied to human activity recognition.

The proposed method uses the totally new approach for features extraction from the sequences of images .The method describes about the recognition of human activity with the help of change in kinetic energy produced by motion of the lattices which in turn based on connected pixels in an image. Then we used the support vector machine as the classifier. The various techniques which we explained above are either lack in the real time implementation of the technique or in qualitative decision making. The proposed technique takes care of both and shows better results.

Chapter-3

RESEARCH, AIMS AND OBJECTIVES

The main focus of this research was to develop an automated video surveillance system. Video surveillance has been an active research topic in these days. Recently our society has faced a lot of terrorist attacks in which a lot of people has been killed. On the border of the country, we have lost many soldiers due to manual monitoring. On the traffic road, road casualties have increased recently. These may be reduced by an effective surveillance system. These are some areas where video surveillance is demand of time.

For automated video surveillance an effective algorithm is needed for detecting any abnormal activity in the video in real time. We have developed a good algorithm for detecting any activities in the video. We have applied this algorithm on three different activities (Walking, jumping and standing position).

Suppose we want to monitor the activities happening in the restricted area.One option is that we appoint a person at the place where CCTV camera is placed. But it is not a good option as it is consuming a lot of manpower. Other option is that we take help from an automated video surveillance system. By analyzing the output of this system, we can find out what is happening at that place. But for effective operation of this option, video surveillance system should operate in real time.

We have tested our algorithm on similar situation. There may be three types of behaviour in the Walking, Jumping and Standing position. Our method is able to recognize these activities in real time.

Chapter-4

METHODOLOGY

We have divided our method in to three parts.

1. Feature extraction
2. Feature representation and Description
3. Classifier

The whole process of activity recognition, using proposed method, can be shown with the help of following block diagram .

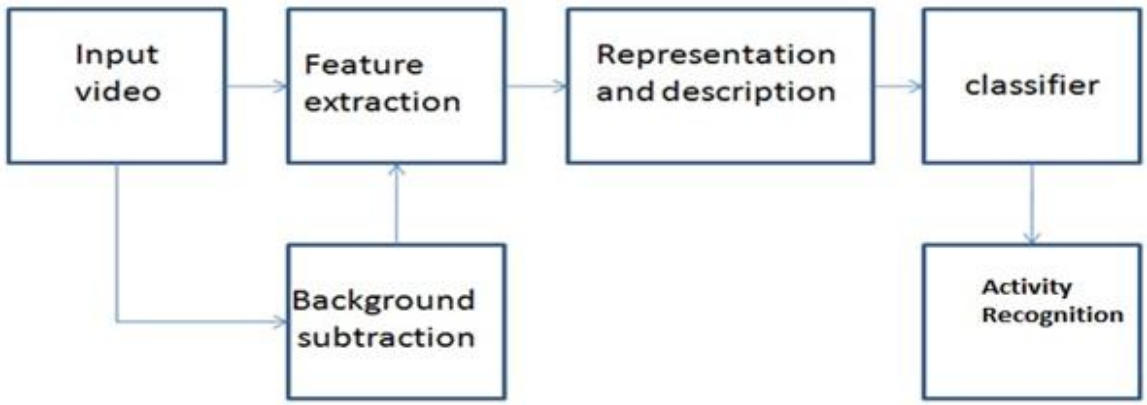


Figure 1: Block diagram of activity recognition system

The steps proposed in the block diagram, as shown in figure 1, are explained in following subsections.

4.1 Pre-processing

There will be need of activity database for training. For this purpose videos of different activities (walking, jumping and stand position) have been taken shown in figures 2(a), 2(b) and 2(c) respectively.

Figure 2(a). "Walking"

Figure 2(b). "Running"

Figure 2(c). "Jumping"

To recognize any particular activity we have segmented the human body into three different parts i.e. lower part(leg),middle part(hand),upper part(head) as shown in figures 3(a), 3(b) and 3(c) respectively.

Figure 3(a). Lower body part (Leg) for different activities

Figure 3(b). Middle body part for different activities

Figure 3(c). Upper body part for different activities

The segmentation process can be done either by manual cropping of the images or by some existed automatic segmentation methods[26].Further more pre-processing is required because there is a large change in the energy due to gray level intensity of the pixel as compared to change in energy due to position.the energy due to intensity value will add those energy which is not of our interest because to recognize activity we are only concentrating on the motion of the connected pixels in sequences of frames. So to compensate this energy effect we removed the background from background subtraction.

4.2 Background subtraction

Background subtraction is a widely used method in Computer Vision for separating or segmenting out the foreground objects from the background of a video. The foreground objects are defined to be the parts of the image that changes and the background is made out of the pixels that stay relatively constant.

In the Computer Vision field, background subtraction is considered to be a low level processing task. It is usually performed as a pre-processing step before more high level tasks such as blob detection; tracking and object detection are performed.

Commonly used techniques for Background Subtraction Include

- Subtraction of reference background frame from each frame.
- Frame Differencing
- Gaussian Mixture Models (GMM)

In the first category, a reference frame is taken as background. Now this frame is subtracted from each frame of the video. Blobs found after background subtraction indicate foreground and remaining portion as background. But this method is not useful for moving background. To remove this problem, we use temporal frame differencing for background subtraction. In this approach consecutive frames are subtracted from each other. This approach is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels, e.g., there may be holes left inside moving entities. Third approach is based on Gaussian Mixture Model. This is highly useful in case of modelling adaptive background. This tackles the problem of moving background and change in illumination of the scene. A brief overview of GMM has been given below.

Gaussian Mixture Model

GMM based method was first introduced by Stauffer and Grimson in 1999, and now it is the most widely used method for background subtraction due to its speed, simplicity and the ease of implementation. In this method, each pixel is modelled as a mixture of Gaussian distributions and any pixel intensity value that does not fit into one of the modelled Gaussian distributions is marked as a foreground pixel.

The background subtraction involves two different tasks, each of which needs to be performed real-time, with having only the video frames as the input.

1. Learning the background model
2. Classifying pixels as background or foreground

Learning the Background Model

Following parameters of each Gaussian component need to be learned dynamically

- The parameters of Gaussians
  - Mean
  - Variance and
  - Weight
- Number of Gaussians per pixel

The update equations for the Gaussian parameters are given below. These equations are executed for each Gaussian component for each pixel at the arrival of each video frame.

$$\begin{aligned}\hat{\pi}_m &\leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \\ \widehat{\vec{\mu}}_m &\leftarrow \widehat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m \\ \widehat{\sigma}_m^2 &\leftarrow \widehat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \widehat{\sigma}_m^2)\end{aligned} \quad (1)$$

Classifying Pixels

$\vec{x}^{(t)}$ = value of a pixel at time t in RGB colour space.

Bayesian decision R – if pixel is background (BG) or foreground (FG):

$$R = \frac{p(BG|\vec{x}^{(t)})}{p(FG|\vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)}|BG)p(BG)}{p(\vec{x}^{(t)}|FG)p(FG)} \quad (2)$$

Initially set p(FG) = p(BG), therefore if the following condition is true we decide that the pixels is a background pixel

$$p(\vec{x}^{(t)}|BG) > c_{thr} \quad (3)$$

But we have used second approach 'frame differencing' for background subtraction because for real time application GMM based approach is not a better option. GMM based approach makes the method very slow as in this method each pixel is modelled by a group of Gaussians.

Now we converted these frames from gray to binary. So in this binary image object is represented by the white pixel as foreground and background by black pixels. Now we applied our method on binary image where we are only concentrating on white pixels which represent the object. So the processing time gets decreases and method becomes well applicable on real time application.

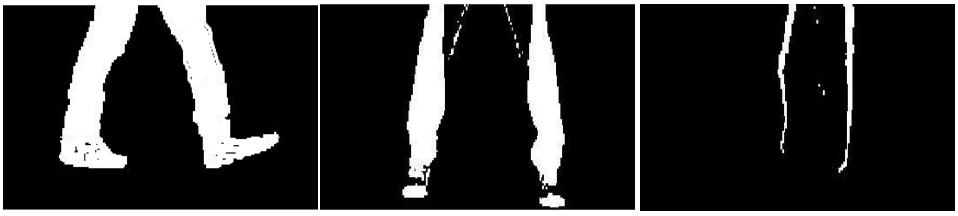


Figure 4(a). Binary image for lower body part for different activities

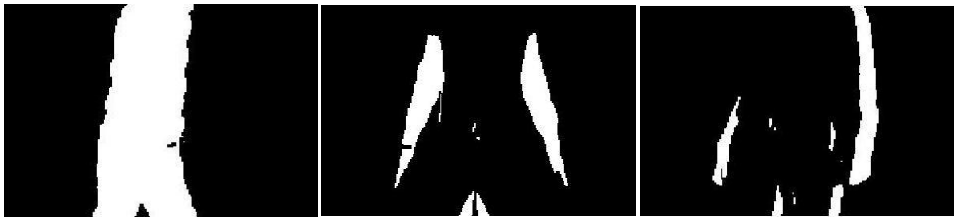


Figure 4(b). Binary image for middle body part for different activities

Figure 4(c). Binary image for upper body part for different activities

Figure 4(a), 4(b) and 4(c) shows the binary images of lower, middle and upper body parts respectively for different activities.

4.3 Features Extraction:

In the image processing, feature extraction from the images is very critical step for developing the method for any application. Extracted features vary from application to application. Basic image processing tools like morphology, histogram, filters etc. are used for feature extraction. Area, height, shape, diameter, centroid of the blob, perimeter of the shape are some examples of the features. Colour information in the image is also an important feature. Efficiency of the method will improve as the number of features increases. But while making a real time system, we have to compromise with the efficiency. For real time application, redundant features are reduced using a dimension reduction technique. Principal Component analysis (PCA) is used for such an application. It is an unsupervised learning and is a standard technique commonly used for data reduction in statistical pattern recognition and signal processing. In this approach to perform dimensionality reduction on some input data, we compute the eigen values and eigen vectors of the correlation matrix of the input data vector, and then project the data orthogonally on to the subspace spanned by the eigen vectors belonging to the dominant eigen values.

Various approaches have been proposed for activity recognition in the video. They can be broadly categorized according to the type of feature extraction and representation adopted. It may be classified in to two ways.

4.3.1 Approach

In the image processing, feature extraction from the images is very critical step for developing the method for any application.

1. The method use background subtracted images of a video. This is done using frame difference method between the successive images of a video.
2. Then we find out the gradient points in every image. This means that we compute the coordinates that represent the gradient in an image whether from black to white or from white to black. These coordinates are then made input for the partial differential equations.
3. The equations are commonly used in mechanics for stress and strain applications but we will be using them for Image Processing in activity recognition. The partial differential equations are used in mechanics for finding out the stress or deformation in the material bodies.
4. In our analysis we will be using them to find out the deformation i.e. displacement of the pixels and obtained values will give the desired feature vector set.

FEM or Finite element method analysis:

The **finite element method (FEM)** (its practical application often known as **finite element analysis** (FEA)) is a numerical technique for finding approximate solutions of partial differential equations (PDE) as well as integral equations. The solution approach is based either on eliminating the differential equation completely (steady state problems), or rendering the PDE into an approximating system of ordinary differential equations, which are then numerically integrated using standard techniques such as Euler's method, Runge-Kutta, etc.

In solving partial differential equations, the primary challenge is to create an equation that approximates the equation to be studied, but is numerically stable, meaning that errors in the input and intermediate calculations do not accumulate and cause the resulting output to be meaningless. There are many ways of doing this, all with advantages and disadvantages.

The finite element method is a good choice for solving partial differential equations over complicated domains (like cars and oil pipelines), when the domain changes (as during a

solid state reaction with a moving boundary), when the desired precision varies over the entire domain, or when the solution lacks smoothness. For instance, in a frontal crash simulation it is possible to increase prediction accuracy in "important" areas like the front of the car and reduce it in its rear (thus reducing cost of the simulation).

Another example would be in Numerical weather prediction, where it is more important to have accurate predictions over developing highly nonlinear phenomena (such as tropical cyclones in the atmosphere, or eddies in the ocean) rather than relatively calm areas.

Finite Element Analysis (FEA) is a tool used for the evaluation of structures and systems, providing an accurate prediction of a component's response subjected to thermal and structural loads. Structural analyses include all types of steady or cyclic loads, mechanical or thermal. Thermal analyses include convection, conduction, and radiation heat transfer, as well as various thermal transients and thermal shocks.

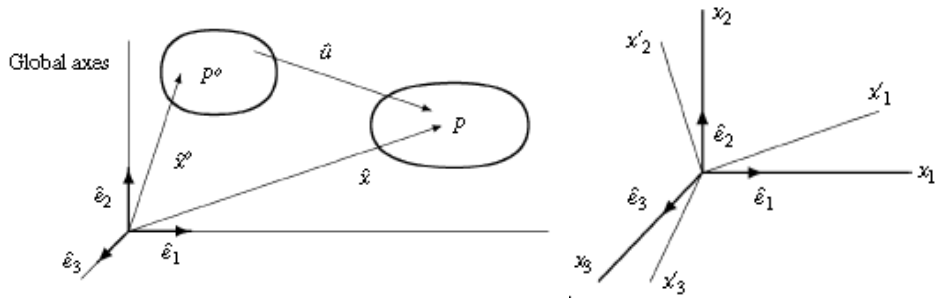
FEA is used to analyze complex geometries, whereas very simple ones (for example, a beam) can be analyzed using hand calculations. For a structure subjected to a load condition (thermal, mechanical, vibratory, etc.) its response (deflection, stress, etc.) can be predicted and measured against acceptable defined limits. In the most simplest terms, this is a factor of safety, which is the ratio of the stress in a component, to the allowable stress of the material. If a factor of safety is too small, the possibility of failure becomes unacceptably large; on the other hand, if the factor is unneccesarily large, the result is a uneconomical or non-functional design. For the majority of structural and machine applications, factors if safety are specified by design specifications or codes written by committees of experienced engineers, such as the American Institute of Steel Construction (design & construction of structural steel for buildings) and the American Concrete Institute (building codes requirements for reinforced concrete).

FEA was largely developed in the 1950's by aerospace engineers to design better aircraft structures. Since then, aided by the rapid growth of computing power, the method has continually developed, and is now the tool of choice for technical analysis by mechanical, civil, biomechanical, and other engineers and this paper is applying it in Image Processing.

The analysis is done by modelling the structure into thousands of small pieces (finite elements). Breaking the entire structure into such small pieces or "elements" is called

Discretization. The solution to the governing equations is closely approximated within each element, resulting in a number of equations that need to be solved for every element. However, each element interacts with its neighbours, i.e., each element's response tightly depends on that of its neighbours, and the responses of their neighbours to those of other neighbours, and so forth. Thus, the element equations cannot be solved alone to render the solution over each element. Instead, all the equations from all the elements over the entire structure need to be solved simultaneously. This task can only be performed by computers. It is noteworthy that, as the structure is broken into a larger number of elements, a greater number of simultaneous equations need to be solved. Thus, typically, results for more complex structures require more computing power.

Now,we tried a new technique of relating this discretization with the coordinate positions in the images of a video.This technique in image processing will measure the discrete positions of the pixels in the image.The successive movements of the pixels from one image to another image in a video will be measured using the following equations.As soon as the strain measures described below are known we will follow the mentioned technique to obtain a feature vector set.



Coordinate descriptions.

5(a) Displacement from an undeformed to a deformed configuration.5(b) Base vectors and rotated coordinate system.

InitialFrame,

$$x = \sum_i x_{i(initialframe)} e_i$$

NextFrame,

$$x = \sum_i x_{i(nextframe)} e_i$$

where both vectors are referred to the common set of unit vectors . The variables x are called the *Lagrangian* and *Eulerian* variables, respectively.

A displacement is the shortest distance travelled when a body moves from one location to another that

$$u = \sum_i e_i - \sum_i x_{i(nextframe)} e_i$$

Let two material points before deformation have the coordinates

$x_{i(initialframe)}, y_{i(initialframe)}$ and

after deformation have the coordinates $x_{i(nextframe)}, y_{i(nextframe)}$. *Then*,

The initial distance between these neighboring points is given by

$$(dS_{initialframe})^2 = \sum_i dx_i^{initialframe} dx_i^{initialframe}$$

and the final distance between the points by

$$(dS_{nextframe})^2 = \sum_i dx_i^{nextframe} dx_i^{nextframe} = \sum_{i,j,m} dx_m / dx_i^{initialframe} dx_m / dx_j^{initialframe} * dx_i^{initialframe} * dx_j^{initialframe}$$

where dx_m / dx_j is called the deformation gradient and

Thus

Strain Measure is:

$$E_{ij} = (dS_{nextframe})^2 - (dS_{initialframe})^2$$

$$= 1/2[\partial u_i / \partial y_{j(initialframe)} + \partial u_j / \partial x_{i(initialframe)} + \sum_m (\partial u_m / \partial x_{i(initialframe)} * \partial u_m / \partial y_{j(initialframe)})]$$

where,

$$x_{m(nextframe)} = x_{m(initialframe)} + u_m$$

$$dx_m / dx_{i(initialframe)} = \partial u_m / \partial x_{i(initialframe)} + \delta_{im}$$

Strain Measures

As a body deforms or in other words it moves or performs any activity, various points in it will translate and rotate. Strain is a measure of the "stretching" of the material points within a body; it is a measure of the relative movement.

The equation (1) is lagrange's differential equation and yields values of strain measure as shown below and then a matrix E is obtained.

We can simply compute the Eigen values for this matrix thereby giving three roots or possible values λ: λ(1), λ(2), λ(3).

These are called the Eigen values or principal values.

Equation (2) helps to find the Strain Energy.

ation (3) helps to find the Potential Energy

Mass has been assumed to be unity for simplicity.

4. We obtain entire features set from the equations-three eigen values,strain energy and potential energy.This feature vector set will be given as input into the classifier.

5. Once a feature selection or classification procedure finds a proper representation, a classifier can be designed using a number of possible approaches. In practice, the choice of a classifier is a difficult problem and it is often based on which classifier(s) happen to be available, or best known, to the user. The simplest and the most intuitive approach to classifier design is based on the concept of similarity: patterns that are similar should be assigned to the same class. So, once a good metric has been established to define similarity, patterns can be classified by template matching or the minimum distance classifier using a few prototypes per class. The choice of the metric and the prototypes is crucial to the success of this approach.

1. $E_{ij} = 1/2[\partial u_i / \partial y_{j\,(lastrefrenceframe)} + \partial u_j / \partial x_{i\,(lastrefrenceframe)} + \sum_m (\partial u_m / \partial x_{i\,(lastrefrenceframe)} * \partial u_m / \partial y_{j\,(\text{lastrefrenceframe})})]$

where

E_{ij} is the strain measure and

$\partial u_i / \partial x_j$ is the deformation gradient in the x-direction

$\partial u_i / \partial y_j$ is the deformation gradient in the y-direction

2. $StrainEnergy = (\sum_{i,j} E_{ij} * C) / 10000$ units

3. $PotentialEnergy = \sum_i u_i$ units

FEATURE VECTOR=$\{\lambda(1),\ \lambda(2),\ \lambda(3),$ STRAIN ENERGY,POTENTIAL ENERGY$\}$

For example:

FOR WALKING:

7.1627 -0.490 -0.0267 3.5679 7;

5.2344 -0.420 -0.028 2.7321 6;

4.3554 -0.1596 -0.0289 2.1211 5;

2.9891 -0.1503 -0.0265 1.5821 4;

6.799 -0.5225 -0.0531 3.5992 8;

FOR JUMPING:

4.2505 -0.351 -0.009 2.1067 3;

4.985 -0.640 -0.008 2.5906 4;

4.2971 -0.460 -0.009 2.2171 4;

5.0951 -0.770 -0.0114 2.7309 5;

3.5969 -0.285 -0.0126 1.8574 4;

Chapter- 5

CLASSIFIERS

5.1 Classifier –overview

Once a feature selection or classification procedure finds a proper representation, a classifier can be designed using a number of possible approaches. In practice, the choice of a classifier is a difficult problem and it is often based on which classifier(s) happen to be available, or best known, to the user. The simplest and the most intuitive approach to classifier design is based on the concept of similarity: patterns that are similar should be assigned to the same class. So, once a good metric has been established to define similarity, patterns can be classified by template matching or the minimum distance classifier using a few prototypes per class. The choice of the metric and the prototypes is crucial to the success of this approach.

We have given a brief introduction of some common classifiers used in pattern recognition and machine learning. These are described below:

5.1.1 k-Nearest Neighbors algorithm

In pattern recognition, the *k*-nearest neighbors' algorithm (*k*-NN) is a method for classifying objects based on closest training examples in the feature space. *k-NN* is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors', with the object being assigned to the class most common amongst its *k* nearest neighbors' (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor.

The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbors'. It can be useful to weight the contributions of the neighbors', so that the nearer neighbors' contribute more to the average than the more distant ones. (A common weighting scheme is to give each neighbor a weight of 1/*d*, where *d* is the distance to the neighbor. This scheme is a generalization of linear interpolation.)

The neighbors' are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data.

Nearest neighbor rules in effect compute the decision boundary in an implicit manner. It is also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity.

5.1.1.1 Algorithm

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, *k* is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the *k* training samples nearest to that query point. Usually Euclidean distance is used as the distance metric; however this is only applicable to continuous variables. In cases such as text classification, another metric such as the overlap metric (or Hamming distance) can be used. Often, the classification accuracy of "k"-NN can be improved significantly if the distance metric is learned with specialized algorithms such as e.g. Large Margin Nearest Neighbors or Neighborhood components analysis. A drawback to the basic "majority voting" classification is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the *k* nearest neighbors' when the neighbors' are computed due to their large number. One way to overcome this problem is to weight the classification taking into account the distance from the test point to each of its *k* nearest neighbors'.

5.1.1.2 Parameter selection

The best choice of *k* depends upon the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when *k* = 1) is called the nearest neighbor algorithm. The accuracy of the *k*-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes. In binary (two class) classification problems, it is helpful to choose *k* to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal *k* in this setting is via bootstrap method.

5.1.2 Neural network

Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. Neural network models attempt to use some organizational principles (such as learning, generalization, adaptivity, fault tolerance and distributed representation, and Pattern Recognition Models computation) in a network of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs. The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data.

The most commonly used family of neural networks for pattern classification tasks is the single-layer perceptron, where the separating hyper plane is iteratively updated as a function of the distances of the misclassified patterns from the hyper plane. If the sigmoid function is used in combination with the MSE criterion, as in feed-forward neural nets

(also called multilayer perceptrons), the perceptron may show a behaviour which is similar to other linear classifiers. It is important to note that neural networks themselves can lead to many different classifiers depending on how they are trained. While the hidden layers in multilayer perceptrons allow nonlinear decision boundaries, they also increase the danger of overtraining the classifier since the number of network parameters increases as more layers and more neurons per layer are added. Therefore, the regularization of neural networks may be necessary. Many regularization mechanisms are already built in, such as slow training in combination with early stopping.

The other most commonly used family of neural networks for pattern classification tasks is the feed-forward network, which includes multilayer perceptron and Radial-Basis Function (RBF) networks. These networks are organized into layers and have unidirectional connections between the layers. Another popular network is the Self-Organizing Map (SOM), or Kohonen-Network, which is mainly used for data clustering and feature mapping. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification/clustering task.

The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge (relative to model-based and rule-based approaches) and due to the availability of efficient learning algorithms for practitioners to use. Neural networks provide a new suite of nonlinear algorithms for feature extraction (using hidden layers) and classification (e.g., multilayer perceptrons). In addition, existing feature extraction and classification algorithms can also be mapped on neural network architectures for efficient (hardware) implementation. In spite of the seemingly different underlying principles, most of the well known neural network models are implicitly equivalent or similar to classical statistical pattern recognition methods. Most NNs conceal the statistics from the user. Despite these similarities, neural networks do offer several advantages such as, unified approaches for feature extraction and classification and flexible procedures for finding good, moderately non linear solutions.

5.1.3 Bayes classifier

The second main concept used for designing pattern classifiers is based on the probabilistic approach. The optimal Bayes decision rule (with the 0/1 loss function) assigns a pattern to the class with the maximum posterior probability. This rule can be modified to take into account costs associated with different types of misclassifications. For known class conditional densities, the Bayes decision rule gives the optimum classifier, in the sense that, for given prior probabilities, loss function and class-conditional densities, no other decision rule will have a lower risk (i.e., expected value of the loss function, for example, probability of error). If the prior class probabilities are equal and a 0/1 loss function is adopted, the Bayes decision rule and the maximum likelihood decision rule exactly coincide.

5.1.4 Hidden markov model

Hidden markov model is a commonly used classifier in pattern recognition and machine learning. It is commonly used for speech processing. Before discussing it, it is needed to know about markov processes:

Consider a system which may be described at any time as being in one of a set of N distinct states S_1, $S_{2,\ldots,}$ S_N as illustrated in figure 6.

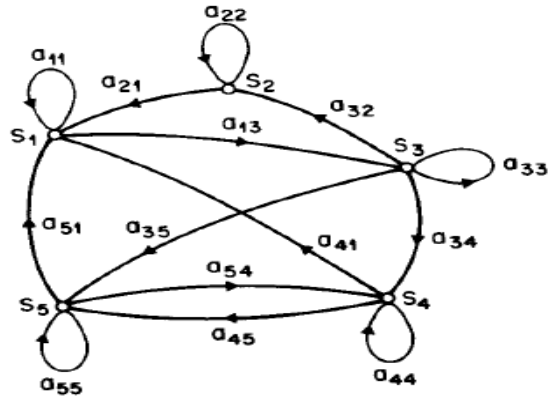


Figure 6, A Markov chain with 5 states (labelled S_1, to S_5) with selected state transitions.

At regularly spaced discrete times, the system undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state. We denote the time instants associated with state changes as $t = 1, 2, \ldots$, and we denote the actual state at time t as qr. A full probabilistic description of the above system would, in general, require specification of the current state (at time t), as well as all the predecessor states. For the special case of a discrete, first order, Markov chain, this probabilistic description is truncated to just the current and the predecessor state, i.e.

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \cdots] = P[q_t = S_j | q_{t-1} = S_i]. \quad (7)$$

The above stochastic process could be called an observable Markov model since the output of the process is the set of states at each instant of time, where each state corresponds to a physical (observable) event.

5.2 Support vector machine

Support vector machines (SVMs) are a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data, and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier. Since an SVM is a classifier, then given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression or

other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite dimensional space, it often happens that in that space the sets to be discriminated are not linearly separable. For this reason it was proposed that the original finite dimensional space be mapped into a much higher dimensional space presumably making the separation easier in that space. SVM schemes use a mapping into a larger space so that cross products may be computed easily in terms of the variables in the original space making the computational load reasonable. The cross products in the larger space are defined in terms of a kernel function which can be selected to suit the problem. The hyper planes in the large space are defined as the set of points whose cross product with a vector in that space is constant. The vectors defining the hyper planes can be chosen to be linear combinations with parameters of images of feature vectors which occur in the data base. With this choice of a hyper plane the points x in the feature space which are mapped into the hyper plane are defined by the relation: Note that if becomes small as grows further from , each element in the sum measures the degree of closeness of the test point to the corresponding data base point . In this way the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated.

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The dominating approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Each of the problems yields a binary classifier, which is assumed to produce an output function that gives relatively large values for examples from the positive class and relatively small values for examples belonging to the negative class. Two common methods to build such binary classifiers are where each classifier distinguishes between (i) one of the labels to the rest (one-versus-all) or (ii) between every pair of classes (one-versus-one). Classification of new instances for one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class (it is important that the output functions be calibrated to

produce comparable scores). For the one-versus-one approach, classification is done by max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, and then the vote for the assigned class is increased by one vote, and finally the class with most votes determines the instance classification.

Now we have the clusters points corresponding to every activity in three dimensions. The best way to classify such data is by Support Vector Machine. At first we need to train the SVM. SVMs can be trained by supervised learning.

As SVM is dedicated to binary classification problems, three popular strategies have been proposed to apply it to multi-class problems. Suppose we are dealing with a *K*-class problem. Thus *K* binary SVMs need to be trained. The scheme is the *one-against-one* method [28, 29], which trains $K\,(K-1)/2$ binary SVMs, each of which discriminate two of the *K* classes. Other newest schemes are Binary Decision Tree (BDT) SVM [30], Directed Acyclic Graph (DAG) SVM [31], which are more complex than other methods. The binary SVM separates the clusters in classes by hyper planes.

We used one against all SVM classification method, it gives satisfactory results. There is one more important issue is selection of the kernel function for SVM. Linear kernel may give erroneous results, instead of that polynomial or radial basis function kernel gives better results.

5.3 Decision:

In the previous section we classified the activity performed by different body parts of a person .the classifier had classified the activity into three different class (walk, jump and stand) for lower, middle and upper body parts; now we need to take a decision for the overall activity performed by a person. For a particular activity like walking, jumping or standing can be recognized if it comes under the class of that activity.

Chapter- 6

RESULT & CONCLUSION

6.1 Experimental Result

In this section experiments have been performed on the test videos. In these videos same activities have been performed for which our system is trained. The mean, variance and bandwidth are found for the test video using the above explained procedure. Finally the data is tested with classifier for its respective class and the overall performance of the system for these videos is obtained.

Figure 7(a)

Figure 7(b)

Figure 7(c)

Figure 7 Test video#1

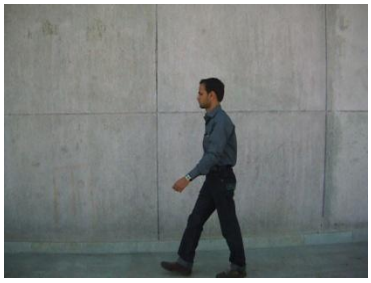


Figure 8(a)

Figure 8(b)

Figure 8(c)

Figure 8 Test video#2

The above figures i.e. figure 9 and figure 10. are the test video frames in which (a) represents walking , (b) represents standing and (c) represents jumping .

Database	Test Video#1	Test Video#2	Total
Recognition Rate	85.3 %	90.7 %	90.0 %

Table 1. Experimental results of Test videos.

We tested the proposed method with different type of kernel function for SVM. The kernel function may be either of linear, quadratic, polynomial or radial basis function type. Recognition rate varies for different kernel functions. As shown in the following table, RBF kernel gives best recognition rate.

SVM type	Linear	Quadratic	RBF
Recognition Rate	91.2 %	92.9 %	90.2 %

Table 2. Recognition rates (in percentage) taking different type of SVM kernel functions

The comparison of recognition rate of proposed method with other conventional methods is given in the following table:

Method	Recognition rate on ORL database
Bayes Network (BN)	80.5
Radial Basis Function Network (RBF)	81.5
Support Vector Machine (SVM)	90.5

Table 3. Comparison of recognition rate (in percentage) with other techniques

Here naive bayes network(NBN), bayes network (BN),multilayer perceptron(MLP),radial basis function(RBF) and support vector machine(SVM) are the methods of behaviour recognition where support vector machine has given best recognition rate (96.3%) among them.

6.2 Conclusion

In this thesis, we have combined the concept of mathematics, mechanics along with signal processing to model a new method for recognizing the activities in the video in real time. This is called Finite Element Method.

Then SVM is used as the classifier.Experimental results demonstrate the effectiveness of our proposed method.

[1] P. Harmo, T. Taipalus, J. Knuuttila, J. Wallet and A. Halme, "Needs and Solutions-Home Automation and Service Robots for the Elderly and Disabled,"*Int'l Conf. Intelligent Robot, Systems*, pp. 3201- 3206, 2005.

[2] J. K. Aggarwal and Qin Cai. Human motion analysis: A review. Computer Vision and Image Understanding, pages 428{440, 1999.

[3] J.K. Aggarwal, Qin Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: Articulated and elastic motion. Computer Vision and Image Understanding, pages 142{156, 1997.

[5] Douglas Ayers and Mubarak Shah. Recognizing human action in a static room. In Proceedings Computer Vision and Pattern Recognition, pages 42{46, 1998.

[5] A. Bobick and J. Davis. Appearance-based motion recognition of human actions. Master's thesis, Massachusetts Institute of Technology, 1996.

[6] Qin Cai and J.K. Aggarwal. Automatic tracking of human motion in indoor scene across multiple synchronized video streams. International Conference on Computer Vision, 1998.

[7] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. IEEE International Conference on Computer Vision Proceedings of the 5th International Conference on Computer Vision, pages 624{630, 1995.

[8] James Davis and Aaron Bobick. The representation and recognition of action using temporal plates. In Proceedings Computer Vision and Pattern Recognition, pages 928{934, 1997.

[9] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. New York Wiley, 1973.

[10] Stephen S. Intille, James Davis, and Aaron Bobick. Real time closed world tracking. In Proceedings IEEE International Conference on Computer Vision and Pattern Recognition, pages 697{703, 1997.

[11] J. Rehg and T. Kanade. Model based tracking of self-occluding articulated objects. IEEE International Conference on Computer Vision Proceedings of the 5th International Conference on Computer Vision, pages 612{617, 1995.

[12] K Rohr. Towards model-based recognition of human movements in image sequences. CVGIP, Image Understanding, 59 n:94{115, 1994.

[13] Nuria Oliver Barbara Rosario and Alex Pentland. A bayesian computer vision system for modeling human interactions. Proceedings of ICVS99, 1999.

[14] Saad Ahmed Sirohey. Human face segmentation and identification. Master's thesis, University of Maryland, 1993.

[15] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time sequential images using hidden markov models. In Proceedings of IEEE Interantional Conference on Computer Vision and Pattern Recognition, pages 379{385, 1992.

[16] Jie Yang, Yangsheng Xu, and Chiou S. Chen. Human action learning via hidden markov model. IEEE Transactions on Systems, Man and Cybernetics, A:34{ 44, 1997

[17] I. Hariataoglu, D. Harwood, L. Davis,"W4: A Real-Time Surveillance of People and Their Activities", *IEEE Transactions on Pattern Analysis and Machine Intelligence* , Vol. 22, No. 8, August 2000.

[18] J.W. Davis, "Sequential Reliable-Inference for Rapid Detection of human actions", IEEE Conf. on Advance Video and Signal Based Surveillance, pp. 169-176, July 21-22, 2003, Miami, FL.

[19] Somboon Hongeng, Ram Nevatia, Francois Bremond, "Video-based event recognition: activity representation and probabilistic recognition methods", *Computer Vision and Image Understanding* 96, (2004) 129-162 .

[20] K. Kira and L. Rendell, "A practical approach to feature selection", *Pro. 9th Int. Workshop on Machine Learning*, 1992, (pp. 249-256).

[21] Schrodinger, E. (1926). *"An Undulatory Theory of the Mechanics of Atoms and Molecules"* (http:/ / home. tiscali. nl/ physis/ HistoricPaper/ Schroedinger/ Schroedinger1926c. pdf). *Physical Review* 28 (6): 1049–1070. doi:10.1103/PhysRev.28.1049.

[22] Tao Zhao, Ram Nevatia, "Tracking Multiple Humans in Complex Situations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, September 2004.

[23] R. Nevatia, F. Lv and T. Zhao, "Self-calibration of a camera from video of a walking human", International Conference on Pattern Recognition, pp. I: 562-567, Quebec City, Canada, August 2002.

[24] Rmer Rosales, Stan Sclaroff, "3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, June 1999.

[25] N. Vlassis and A. Likas, "The kurtosis-EM algorithm for Gaussian mixture modelling," IEEE Trans. SMC ,1999.

[26] Werghi.N,Yijun Xiao,Siebrt,J.P., "A functional based segmentation of human body scans in arbitrary postures, IEEE transections on Cybernatics, Part:B, Volume. 36, Issue. 1, Pub: 2006, PP. 153-165.

[27] Altman, Naomi and Christian Leger (1995): "Bandwidth selection for kernel distribution function,estimation," Journal of Statistical Planning and Inference, 46, 195-214.

[28] S. Knerr, L. Personnaz, and G. Dreyfus, *Nurocosingle-layer learning revisited: A stepwise procedure for building and training a neural network*, Springer, 1990.

[29] U. Kressel, *Pairwise classification and support vector machines*, in Advances in Kernel Methods- Support Vector Learning (1999).

[30]Hossam Osman.*Novel multiclass SVM-based binary decision tree classifier*. IEEE International Symposium on Signal Processing and Information Technology, pages 880-883, 2007.

[31]Platt, J., Cristanini, N., Shawe-Taylor, J.: *Large margin DAGs for multiclass classification.* Advances in Neural Information Processing Systems 12. MIT Press. Pages 543–557, 2000.

[32] Azzalini, A. (1981): "A note on the estimation of a distribution function and quantiles by a kernel method," Biometrika, 68, 326-328.

[33] Jin, Zhezhen and Yongzhao Shao (1999): “On kernel estimation of a multivariate distribution function,”Statistics and Probability Letters, 41, 163-168.

[34] Fengjun Lv, Jinman Kang, Ram Nevatia, Isaac Cohen, Gerard Medioni “Automatic tracking and labeling of human activities in a video sequence”,IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Prague, May 11-14, 2004.

[35] O. Masoud, N. Papanikolopoulos, “Recognizing human activities”, IEEE Conf. on Advanced Video and Signal Surveillance, 2003.

[36] Aaron F. Bobick, James W. Davis, “The Recognition of Human Movement Using Temporal Templates”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 3, March 2001.

[37] Alexei Efros, Alexander Berg, Greg Mori, Jitendra Malik, “Recognizing Actions at a Distance”, *IEEE International Conference on Computer Vision*, Nice, France, October 2003.

EMRS DTC Technical Conference – Edinburgh 2006

[38] Tao Xiang and Shaogang Gong, “Video Behaviour Profiling and Abnormality Detection without Manual Labelling”, Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05)

[39] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: real-time tracking of the human body,” IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 780–785, July 1997.

[40] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, “Tracking groups of people,” Comput. Vis. Image Understanding, vol. 80, no. 1, pp. 42–56, 2000.

[42] I. A. Karaulova, P. M. Hall, and A. D. Marshall, “A hierarchical model of dynamics for tracking people with a single video camera,” in Proc. British Machine Vision Conf., 2000, pp. 262–352.

[43] S. Ju, M. Black, and Y. Yaccob, “Cardboard people: a parameterized model of articulated image motion,” in Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition, 1996, pp. 38–44.

[44] S. A. Niyogi and E. H. Adelson, “Analyzing and recognizing walking figures in XYT,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1994, pp. 469–474.

[45] S. Wachter and H.-H. Nagel, “Tracking persons in monocular image sequences,” Comput. Vis. Image Understanding, vol. 74, no. 3, pp. 174–192, 1999.

[46] N. Paragios and R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 266–280, Mar. 2000.

[47] N. Peterfreund, “Robust tracking of position and velocity with Kalman snakes,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 564–569, June 2000.

[48] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density,” in Proc. European Conf. Computer Vision, 1996, pp. 343–356.

[49] R. Polana and R. Nelson, “Low level recognition of human motion,” in Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects, Austin, TX, 1994, pp. 77–82.

[50] D.-S. Jang and H.-I. Choi, “Active models for tracking moving objects,” Pattern Recognit., vol. 33, no. 7, pp. 1135–1146, 2000.

[51] V. K. Madasu and S. Vasikarla, “Fuzzy edge detection in biometric systems”, in 36th Applied Imagery Pattern Recognition Workshop, IEEE, pp. 139-144, 2007.

[52] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel, “Toward robust automatic traffic scene analysis in real-time,” in Proc. Int. Conf. Pattern Recognition, Israel, 1994, pp. 126–131.

[53] R. Plankers and P. Fua, “Articulated soft objects for video-based body modelling,” in Proc. Int. Conf. Computer Vision, Vancouver, BC, Canada, 2001, pp. 394–401.

[54] D.O. Aborisade, “Fuzzy Logic Based Digital Image Edge Detection”, in Global journal of Computer Science and Technology, pp. 78-84, 2010.

[55] C. Piciarelli, C. Micheloni, and G. Foresti, “Trajectory-based anomalous event detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 11, pp. 1544–1554, 2008.

[56] T. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov, “Computer vision workload analysis: Case study of video surveillance systems,” Intell. Technol. J., vol. 9, no. 2, pp. 109–118, 2005.

[57] W. Hu, T. Tab, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” IEEE Trans. Syst. Man Cybern., vol. 34, no. 3, pp. 334–352, 2004.

[58] P. Kumar, S. Ranganath, H. Weimin, and K. Sengupta, “Framework for real time behavior interpretation from traffic video,” IEEE Trans. Intell. Transp. Syst., vol. 6, no. 1, pp. 43–53, 2005.

[59] M. Bennewitz, G. Cielniak, and W. Burgard, “Utilizing learned motion patterns to robustly track persons,” in Proc. IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Nice, France, 2003, pp. 102–109.

[60] N. M. Oliver, B. Rosario, and A. P. Pentland, “A Bayesian computer vision system for modeling human interactions,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no. 8, pp. 831–843, 2000.

[61] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa, “Shape activity: A continuous state hmm for moving/deforming shapes with application to abnormal activity activity detection,” IEEE Trans. Image Processing, vol. 14, no. 10, pp. 1603–1616, 2005.

[62] N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition,” Image Vis. Comput., vol. 14, no. 8, pp. 609–615, 1996.

[63] N. Sumpter and A. Bulpitt, “Learning spatio-temporal patterns for predicting object behavior,” Image Vis. Comput., vol. 18, no. 9, pp. 697–704, 2000.

[64] C. Stauffer and E. Grimson, “Learning patterns of activity using real-time tracking,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no. 8, pp. 747–757, 2000.

[65] I. Saleemi, K. Shafique, and M. Shah, “Probabilistic modeling of scene dynamics for applications in visual surveillance,” IEEE Trans. Pattern Anal. Machine Intell., vol. 31, no. 8, pp. 1472–1485, 2009.