JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Suspicious Activity Recognition in Video Sequences

[1]Tripti Singh, [2]Ritu

[1,2]Department Of Computer Science and Engineering, Ganga Institute Of Technology and Management, Kablana
cris.tripti12singh@gmail.com, ritu.cse@gangainstitute.com

Abstract :

The rapid proliferation of video surveillance systems has elevated the importance of detecting suspicious activities in real-time video sequences as a critical component of ensuring public safety and security. This research delves into the domain of Suspicious Activity Recognition in Video Sequences (SARVS), exploring innovative techniques and methodologies to effectively identify and classify suspicious behaviors in video streams. The majority of the currently used video surveillance systems rely on humans or manual examination of the video's content for any suspicious activities. When there is a lot of data to analyse, this strategy is not helpful. Analysis in this situation is often only performed after anything goes wrong. However, the automatic method of analysing and spotting suspicious behavior will aid in swiftly and effectively seeing any such anomalous activity and may even issue an alert before the occurrence of any significant fatality. Three elements make up the fundamental strategy for autonomous video surveillance: recognizing moving objects, following them, and spotting anomalous behavior. Segmenting moving objects from stationary backdrop is the initial stage in the motion detection process. The most widely used approaches for object detection are temporal differencing, background removal, optic flow and statistical approach. Due to dynamic environmental variables including shifting lighting, swaying tree branches, and shadows, object segmentation is challenging and fraught with issues. Therefore, a reliable and quick video surveillance system is required. The next stage in video analysis is tracking, which is basically the idea of temporal congruence between observed motion objects in the frame sequences. This process determines the segmented items' temporal recognition and produces coherent data on the segmented objects in the monitoring area. The data from the tracking stage is often applied to assist motion segmentation, extracting the features technique and inspection of a typical behavior. Recognizing the odd or unusual behavior in a video is the last stage. The output of these algorithms may be utilized to provide the worker with high level semantic data, which will enable operant to make informed judgments.

Keywords – video surveillance, object motion, Gaussian model, Bayesian framework etc**.**

I. INTRODUCTION

Video surveillance systems have become an indispensable tool in ensuring public safety and security in various domains, including transportation, law enforcement, retail, and critical infrastructure. The ability to recognize and respond to suspicious activities in real-time video sequences is paramount for proactive threat prevention and effective incident management. As a result, Suspicious Activity Recognition in Video Sequences (SARVS) has garnered considerable attention from researchers and practitioners in recent years. This thesis delves into the field of SARVS, seeking to develop an innovative and robust system capable of accurately detecting and categorizing suspicious behaviors in video streams.

In the analysis of video sequences while monitoring, frames are observed in search of particular behaviors that are inappropriate or that may indicate the existence of unusual behavior. In order to conduct activity-based analysis of the video sequences, it is essential for video analysis systems to identify activities. A computer can read and comprehend visual data from the outside environment using a set of instructions called a computer vision algorithm. These algorithms can be used for a variety of purposes, including object detection, image classification, facial recognition, and video tracking. They are used to analyse and process images and videos. Finding timely and reliable activity statistics, however, presents major difficulties. Activity detection is extremely important in many applications, particularly in the surveillance sector. The copious amounts of data which is present in video sequences often impede decision-making for computer systems.

The majority of the currently in use video surveillance systems rely on humans or manual examine the video's content for any suspicious activities. When there is a lot of data to analyse, this strategy is not helpful. Analysis in this situation is often only

performed after anything goes wrong. However, the automatic method of analysing and spotting suspicious behavior will aid in swiftly and effectively seeing any such anomalous activity and may even issue an alert before the occurrence of any significant fatality. This involves tracking of the foreground object for detection and predict trajectory and identify suspicious activity [1]. The work of analyzing video sequences by the human operator must be assisted by automatic processing tools due to the volume of data that needs to be processed in new surveillance systems. The human operator will also be able to analyze multiple environment monitoring [2].

"Recognizing the actions occurring in the video is necessary for a number of applications, including content-based video annotation and retrieval, highlight extraction, and video summarization. Analysis of human activity in video has a variety of applications, from security and surveillance to entertainment and personal archiving, and is becoming more and more significant. This topic is difficult to address due to a number of difficulties at different levels of processing, including robustness against mistakes at low-level processing, view- and rate-invariant representations at midlevel processing, and semantic representation of human actions at higher level processing" [7].

Three elements make up the fundamental strategy for autonomous video surveillance: recognizing moving objects, following them, and spotting anomalous behavior. Segmenting moving objects from stationary backdrop is the initial stage in the motion detection process. The most widely used approaches for object detection are temporal differencing, background removal [4], optic flow and statistical approach. Due to dynamic environmental variables including shifting lighting, swaying tree branches, and shadows, object segmentation is challenging and fraught with issues. Therefore, a reliable and quick video surveillance system is required [13, 14].

The next stage in video analysis is tracking, which is basically the idea of temporal congruence between observed motion objects in the frame sequences. This process determines the segmented items' temporal recognition and produces coherent data on the segmented objects in the monitoring area. The data from the tracking stage is often applied to assist motion segmentation, extracting the features technique and inspection of a typical behavior [16]. Recognizing the odd or unusual behavior in a video is the last stage. The output of these algorithms may be utilized to provide the worker with high level semantic data, which will enable operant to make informed judgments.

The thesis proposes a system which helps to solve the technical problem of video analysis for suspicious activity recognition. The input taken by the system is offline pre stored videos. The activity recognized in this system involves detection of two scenarios i.e holding the briefcase or bag or packet and second is leaving the briefcase or bag. There is only one person involved in the frame. We create a statistical background model and remove background information from the video stream to recover foreground objects.

II. LITERATURE REVIEW

Video surveillance systems have become critical for public safety and security in various domains, such as transportation, law enforcement, and public spaces. One of the essential tasks of video surveillance is recognizing suspicious activities that could indicate potential threats or criminal behavior. This literature survey explores the existing research on suspicious activity recognition in video sequences, focusing on different methodologies, algorithms, and datasets employed in the field.

We now provide the survey, which solely covers writings connected to the same area as our thesis topic. Fig. 2.1 depicts a general structure for video surveillance [4, 8, 21].

Diagram 2.1: Generic Framework of the system

Here's a detailed explanation of each step in the flowchart:

- Video:

The process starts with input video data. This could be a recorded video sequence captured by cameras, surveillance footage, or any other source of video data.

- Foreground Item Identification: In this step, the algorithm processes the video frames to identify the foreground items or regions of interest. The foreground items represent the moving or dynamic elements in the scene, typically corresponding to the objects or people of interest.

Foreground item identification can be achieved through background subtraction techniques, which compare each frame to a reference background model to extract moving objects.

- Object Tracking: Once the foreground items are identified, the next step is to track these objects across consecutive frames. Object tracking involves assigning unique identifiers or labels to the objects to maintain their identities as they move.

Various object tracking algorithms, such as correlation filters, Kalman filter, or particle filter, can be used to predict and update the object positions in subsequent frames.

- Classification:After the objects are tracked, the system extracts relevant features from each object to represent its appearance and motion characteristics. These features are used to classify the objects into different predefined categories or classes.

- Activity Detection: Activity detection is the final step in the flowchart, where the system analyzes the sequence of classified objects to recognize human activities or actions. The system looks for patterns in the sequence of object classes and their temporal relationships to identify specific activities or behaviors.

2.1 Background Subtraction

Using the fundamental method of background subtraction, activity identification algorithms can distinguish between moving foreground items in a video sequence—like people or cars—and the stationary background. It is possible to examine and track moving items by separating them, which makes it possible to identify diverse actions. In computer vision and video analysis, the technique of background removal is frequently used to identify moving objects or activities in a video stream. The main objective is to extract the foreground objects from the stationary background, which are then used for subsequent activity recognition tasks. Due to abrupt lighting, repeated movements, cluttering, and occlusion, the accurate method for recognizing moving objects is challenging [14]. Foreground background segmentation approaches that are often employed include frame differencing, optical flow, and statistical methods [23]. Below, we go into further detail about these methods.

2.1.1 Frame Differencing

In order to identify the area of a moving object, temporal differencing compares the differences between pixels in subsequent frames [23]. The difference between the new frame and the background model is computed for each new frame in the video. This distinction draws attention to areas where there has been a substantial change, signifying the existence of moving things. This technique responds very quickly to frequent scene changes. In dynamic situations, it typically fails to identify all of the pixels pertinent to an item.

These fundamental steps are followed by the frame differencing algorithm:

Image acquisition is the process of taking successive frames or images from a video stream or image sequence.

Grayscale Conversion: The frames are frequently converted to grayscale to streamline the processing and concentrate on intensity fluctuations.

Frame Difference Calculation: Subtract the matching pixel value from the preceding frame for each pixel in the current frame. The outcome is a brand-new image known as the difference image that highlights regions with notable changes in pixel intensity.

Thresholding: The difference picture is subjected to a thresholding procedure in order to further separate the moving regions. The foreground or moving items are represented by pixels with values above a predetermined threshold, while the background is represented by pixels with values below the threshold.

Post-Processing: Additional post-processing techniques, such as morphological operations (such as erosion and dilation) or connected component analysis, may be used to improve the segmentation and eliminate noise, depending on the application and the amount of noise in the difference picture.

2.1.2 Optical Flow Based

In computer vision and video analysis, an advanced technique for spotting moving objects in a scene is called optical flow-based background subtraction. Optical flow takes into account the movements of individual pixels over time, as opposed to conventional background subtraction techniques, which only take into account pixel intensity differences between succeeding frames. An initial background model serves as the foundation for the optical flow-based background removal. To depict the static background, use the first few frames or an average of those frames. The optical flow algorithm compares each new frame's pixels to their corresponding pixels in the backdrop model to determine each pixel's velocity vector. The displacement and direction of pixels from the backdrop to the current frame are represented by the motion vectors.

The magnitude of the motion vectors is subjected to a threshold in order to detect moving objects. Foreground (moving objects) pixels are those with motion magnitudes above the threshold, whereas background pixels are those with motion magnitudes below the threshold.In order to identify moving items in a frame, optical flow algorithms employ the flow vectors of objects' movements across time [7]. The relative mobility of objects over several frames is investigated via optical flow sequences based on variations in light and the speed at which things travel. Due to noise and changing light, it is impossible to compute optical flow without utilizing specialized hardware [23].

2.1.3 Statistical Approach

To address the limitations of simpler methods of background removal, statistical approaches are more sophisticated techniques that employ the statistical properties of individual pixels [7]. These statistical techniques preserve and dynamically update background

scene-related pixel data. Each pixel's statistics are compared to those of the background data, and any pixels that deviate from the background are then recognised. Background removal by statistical approaches involves using statistical models to distinguish foreground objects from the background in an image or a sequence of images. The primary goal is to create a binary mask that identifies pixels belonging to the foreground objects, effectively removing the background. A well-liked statistical technique for background modeling and subtraction is GMM.

The backdrop pixel intensities in GMM are expected to follow a concatenation of different Gaussian distributions. The background's various textures or lighting conditions are represented by each Gaussian component.

The Gaussian components' mean, variance, and weight are estimated during the modeling phase using a series of training frames devoid of foreground objects to determine which Gaussian components best match the background pixel intensities.

The pixel intensities are compared to the Gaussian components whenever a new frame is processed. A pixel is categorized as being in the foreground if its intensity considerably differs from the background model. This method is more reliable when dealing with lighting variations, shadows, and noisy situations [23].

2.2 Object Tracking

Finding and following particular items of interest over time as they move and interact inside the video frames is the process of object tracking in a video scene. In order to provide information on the position, size, and movement patterns of the target object(s), object tracking algorithms attempt to preserve the trajectory of the object(s) over a series of frames. In order to improve tracking results and lessen jitter or noisy trajectories, tracking algorithms frequently use filtering and smoothing techniques. As per the requirements of the proposed system, object tracking in a video scene can be divided into several categories. For monitoring an item as a whole, there are two popular methods: correspondence matching [21] and position prediction or motion estimation [17]. The techniques used to find and monitor the human body components use a model-based methodology. The different ways of tracking are as mentioned below:

2.2.1 Model-Based Tracking

A sophisticated method of object tracking in a video scene called "model-based tracking" entails building a dynamic model of the target object's look, motion, and form. Initializing the object model in the first frame is the first step in model-based tracking. Usually, manual annotation or automatic object detection techniques are used to do this. The target object's visual features, such as color, texture, and shape, are represented by the appearance model. New observations from the video frames are continuously used to update the look and motion models. The object's position and orientation in the following frame are then predicted using the modified model. In order to manage changes in object appearance or motion over time, the tracker can modify the appearance and motion models.

Retraining the appearance model or dynamically modifying the motion model parameters can accomplish this. Stick Diagrams, 2-Dimensional contours, and volumetric models may all be used to represent the geometric structure of a person [11]. The specifics of each of these are provided below.

2.2.1.1 Representation using Stick Diagram

Stick diagrams, commonly referred to as stick figures, are an easy-to-understand abstraction of a person's geometric make-up. In a stick diagram, lines or "sticks" are used to represent the human body's limbs and joints. The stick figure concentrates on capturing the fundamental position and movement of a person rather than providing specific information about body proportions, face features, or attire. Stick diagrams are frequently used to depict body movements and positions in instructional materials, animation storyboarding, and motion analysis. The human body is shown using stick Diagrams by using an arrangement of line segments connected by joints [23]. Meghna Singh and others [36] in her work used stick Diagram, which has ten articulated sticks and six joints, was used to illustrate the anatomy of the human body in shadow.

2.2.1.2 Two-dimensional contour

Using 2D shapes or silhouettes, 2-dimensional contours show the geometric make-up of a person. These limits usually show a person's body in a flat manner without capturing its depth or volume. Stick diagrams lack the granularity that 2-dimensional contours may provide, making it easier to understand the proportions and features of the human body. They are simply produced from pictures or motion capture systems that use silhouettes.

Applications for human position estimation, gesture recognition, and silhouette-based motion tracking all make use of contour-based representations. The extrapolation of the human body in the picture plane is strongly related to this kind of human body depiction. Segments of the human body are represented in this way as two dimenstional ribbons [21].

2.2.1.3 Volumetric Models

Volumetric models, which depict the depth and volume of the body, capture the geometric structure of an individual in three dimensions. These models define the body as a set of approximate 3D shapes that represent the body's structure, such as cuboids, spheres, or 3D meshes.

In order to accurately depict, animate, and interact in 3D environments, volumetric models provide a more accurate and detailed portrayal of the human body.

Volumetric models are frequently employed in computer graphics, virtual reality, character animation, as well as in fields like biomechanics, human motion analysis, and medical imaging.

Due to their limitations on camera angle, 2-D models have several drawbacks. With the use of three dimensional models and among others, many researchers are attempting to uncover the geometric structure of the human body in more detail [16].

2.2.2 Region-Based Tracking

Using the idea of regions of interest (ROIs) to follow particular objects across time is known as region-based tracking of objects in an image. In this method, the tracker first locates a region in the first frame that contains the target item. The tracker then updates

and modifies the region in later frames using a variety of approaches to keep track of the object even when it moves or undergoes aesthetic changes. Both 2D image tracking and 3D object tracking frequently employ region-based tracking.

Region-Based When using methods like picture segmentation or region-based object detection, tracking can certainly see the human body as a collection of blobs or areas. A "blob" is a connected area or group of pixels in this context that have similar properties, such as comparable color, texture, or intensity. It is feasible to track and evaluate the movement and attitude of the human body by locating and tracking these blobs or patches in the image.

The head, chest, arms, and legs are just a few examples of the various body parts or segments that can be detected and tracked using region-based tracking techniques in the context of tracking the human body. Every body part is shown as its own distinct blob or zone of interest.

Then, Gaussian distributions are used to simulate the human body as well as the backdrop picture. To efficiently manage shadows while segmenting a moving object, “Javier Varona et al.”[3] devised a backdrop subtraction approach that incorporated gradient information and colour. The tracking procedure is then carried out at other levels of abstraction, such as regions, persons, etc. A bounding box exists for each zone that can combine and separate.

2.2.3 (Snake) Active Contour technique

The Snake, often referred to as the Active Contour model or the Active Snake model, is a potent method for object tracking and picture segmentation in computer vision and image processing. The idea of snakes or flexible outlines that may move and adjust to the edges or boundaries of things in an image served as its inspiration. The Snake model is frequently employed because it can properly represent the contours and shapes of complex objects, even when there is noise and brittle edge. The active contours method, often known as snakes, is an iterative algorithm for segmenting images into expanding regions. By specifying starting curves on an image and using the active contour function to grow the curves towards object boundaries, you may apply the active contour algorithm. Tracking using active contour models immediately determines an object's form. The goal is to dynamically update the boundary contour of the items and depict it. “A variant framework for identifying and following moving objects in a video was addressed by Liang Wang et al.” [22].

2.2.4 Object detection and tracking based on features

Object detection and tracking based on features is a common approach in computer vision that involves using distinctive visual features to detect and track objects in images or video sequences. These features serve as distinctive landmarks or descriptors that enable the identification and localization of objects, even under varying conditions such as changes in appearance, viewpoint, and occlusions. Feature-based object detection and tracking are widely used in various applications, including surveillance, robotics, augmented reality, and human-computer interaction. The tracking framework [22] based on features include feature extraction and feature matching. The first step is to extract relevant features from the images or video frames. These features should be distinctive and robust to changes in lighting, scale, and orientation. Common methods for feature extraction include:

- Finding the image's corner points that correspond to significant features is known as corner detection.
- Blob detection is the process of identifying areas in a picture that have a particular intensity or color distribution.
- Detecting edges and contours in an image is known as edge detection.

The extracted features need to be represented in a way that allows for efficient matching and comparison. Descriptors of common features include:

- The image's local gradient distribution is described by the histogram of oriented gradients (HOG).
- Extraction of scale-invariant keypoints and associated descriptors using the Scale-Invariant Feature Transform (SIFT).
- Speeded-Up Robust Features (SURF): A quicker computing version of SIFT.

For object detection, a model or classifier is trained to recognize specific objects or classes based on the extracted features and their representations. Once an object is detected in the initial frame, its features are used to track the object in subsequent frames. While it is simpler to extricate low-level features however it might be more challenging to monitor higher-level features like blobs and lines. As a result, there is should be a optimal balance between feature richness and tracking efficiency. In their study, “Jiang Dan and Yu Yuan employed point-feature tracking” [38].

2.3 Activity Recognition

The issue of identifying an event from visual sequences naturally arises after effectively monitoring moving objects in a movie from frame to frame. Action identification and description are both involved in activity recognition [7].

The activity recognition method makes the assumption that each item type's form is known [5]. The three fundamental categories of things are people, vehicles, and carrying objects. Either the detection or tracking mechanisms or the system users specify this information [1]. These are all challenging issues that have attracted scientific interest.

2.3.1 FSMs

The temporal sequences of human actions can be modeled and represented using Finite State Machines (FSMs) in activity recognition. FSMs aid in the capturing of dynamic behavior and changes between various activity states during activity recognition based on input events or observations from sensors, video data, or other sources. When dealing with tasks that need a series of smaller tasks or have complicated temporal patterns, FSMs can be especially helpful. FSMs can be used in activity recognition in the following ways:

Activity States:

Each activity is represented as a state in the FSM during activity recognition. Each condition is associated with a distinct activity or sub-activity.

States Transistions:

Specific events, sensor inputs, or modifications in the recorded data can all trigger changes in activity states.As an illustration of the sequential nature of the overall activity, the conclusion of one sub-activity could result in the shift to the following state.

Temporal Sequences:

The FSM depicts the temporal sequences of actions taken by the user or subject by capturing the changes between activity states.

Activity Context and Adaptation:

FSMs can be designed to account for contextual information and adapt to variations in activity patterns or execution styles.

For instance, an FSM for "cooking" might have different transitions based on whether the person is preparing breakfast, lunch, or dinner.

Activity Recognition Decision:

When new sensor data or observations are fed into the FSM, it progresses through different states based on the inputs.The final recognized activity is determined by reaching a certain state or a set of states that correspond to specific activity classes.

FSM formalism has the innate capacity “to capture sequences that enable it to be coupled to other abstractions, including object-based abstraction and pixel-based abstraction” [24]. Due to its simplicity, pedagogy, and capacity to describe temporal sequence, FSMs are a crucial tool in event interpretation.

2.3.2 Bayes Net

Bayesian Networks, also known as Bayes Nets or Belief Networks, are probabilistic graphical models that can be applied in activity recognition to model the dependencies between different variables and events. They are particularly useful when dealing with uncertain or probabilistic relationships between activity-related factors. Bayes Nets enable efficient reasoning and decision-making under uncertainty, making them suitable for activity recognition systems that deal with noisy sensor data, variable activity patterns, and uncertain observations.

Here's how Bayes Nets can be applied in activity recognition:

Modeling Dependencies: In activity recognition, Bayes Nets can represent the dependencies between various factors that influence activities. These factors can include sensor data, environmental conditions, previous activities, and user behavior.

Node Representation: Each node in the Bayes Net represents a variable or factor relevant to activity recognition. For example, nodes could represent sensor readings, activity labels, or contextual information.

Probabilistic Relationships: The connections (edges) between nodes in the Bayes Net represent probabilistic relationships. Conditional probabilities are assigned to these edges, indicating the likelihood of an event or state given the state of its parent nodes.

Observations and Inference: Bayes Nets allow incorporating observed data (e.g., sensor readings) as evidence to update the probability distributions of activity states. By performing probabilistic inference, Bayes Nets can calculate the probability of different activity states given the observed evidence.

Activity Recognition Decision: The Bayes Net can be used to make activity recognition decisions by selecting the most probable activity state based on the evidence.

Dynamic and Temporal Models: Bayes Nets can handle dynamic activity recognition, where activity states change over time, by using temporal extensions like Dynamic Bayesian Networks (DBNs).

In belief Networks, nodes represent the random variables, which can be discrete (a limited number of states) or continuous (defined by a parametric distribution) [20]. The design of Bayesian networks enables the application of conditional independence and joint probability over all variables with minimal parameters [5].

In order to identify events like interior and aerial surveillance, Bayesian networks have been utilised. For example, American football plays and parking lot surveillance have both been recognised using more sophisticated Bayesian networks [33].

2.3.3 HMMs

In order to represent and recognize temporal sequences of activities, Hidden Markov Models (HMMs) are frequently utilized in activity recognition. HMMs are probabilistic graphical models that are capable of handling sequential data and are especially good at capturing the temporal dependencies and fluctuations in activity patterns. They work effectively for tasks requiring the recognition of activities in which the sequence of observations (such as sensor data or video frames) is important for identifying the underlying activity.

HMMs is based on directed graph and statistical Markov model that includes the temporal development of the observed state and it is assumed that the system under study is a Markov process with unobserved (hidden) states. With a single time slice, the concealed state is represented by one variable and observed state by the other [10].

Temporal Sequence Modeling: HMMs are designed to model sequences of observations over time. In activity recognition, these observations can be sensor data, video frames, or any other relevant data that reflects the activities being performed.

Hidden States: HMMs use hidden states to represent the underlying activity classes or behaviors that generate the observed data. Each hidden state corresponds to a specific activity.

Observation Likelihoods: For each hidden state, an observation likelihood is defined, representing the probability of generating the observed data given that the system is in that state.

These observation likelihoods are often modeled using probability distributions such as Gaussian distributions for continuous data or discrete probability distributions for categorical data.

Transition Probabilities: HMMs also have transition probabilities that govern the transitions between hidden states over time. These transition probabilities capture the temporal dependencies between activities and allow the model to learn the likelihood of transitioning from one activity to another.

Learning: Training an HMM involves estimating the model's parameters, including the observation likelihoods and transition probabilities, from labeled training data. The Baum-Welch algorithm (Expectation-Maximization) is commonly used for learning the HMM parameters from the data.

Activity Recognition Decision: Once the HMM is trained, it can be used to recognize activities in new sequences of observations.

2.3.4 Conditional Random Fields (CRFs)

Conditional Random Fields (CRFs) are probabilistic graphical models used for structured prediction tasks, particularly in the context of sequence labeling, image segmentation, and natural language processing. CRFs are a type of discriminative model that models the conditional probability of output labels given input observations, taking into account the dependencies between neighboring output labels. They have become popular in various machine learning and pattern recognition applications due to their ability to model complex dependencies and produce accurate and coherent predictions in structured data.

1. Formulation: CRFs are used for solving supervised learning problems where the output labels are structured, such as sequences or grids, and there are dependencies between neighboring labels.
Given a set of input observations (features) x and a set of output labels (states) y, the goal is to learn the conditional probability P(y | x) over all possible label sequences y.
2. Feature Functions: CRFs represent the input observations and output labels using feature functions. These functions map the input-output pairs (x, y) to real-valued scores, representing the compatibility between the input and output labels. Feature functions can capture both local and global dependencies between output labels, making CRFs suitable for capturing contextual information in structured data.
3. Modeling Dependencies: CRFs model the dependencies between neighboring output labels using pairwise potential functions. These potential functions capture the interaction between adjacent labels in the sequence. The pairwise potentials are often defined based on the feature functions and can be learned from training data.
4. Training: CRFs are typically trained using labeled data, where both input observations and corresponding output labels are available. The training process involves estimating the parameters (weights) of the feature functions to maximize the conditional likelihood of the output labels given the input observations.
5. Inference: Once trained, CRFs can be used for prediction or inference on new, unseen data. The inference process finds the most likely output label sequence (i.e., the sequence with the highest conditional probability) given the input observations and learned model parameters.
CRFs belong to the graphical model of machine learning and part of the family of statistical modeling techniques used for structured prediction and frequently utilized in pattern recognition and machine learning. CRFs are graphical framework which is undirected that generalize the HMM by substituting feature functions for the transition probabilities that correspond to the global observation [23]. It is possible to apply to CRFs the recognized issues with observation and assessment for HMMs. With the use of convex optimization techniques, such as conjugate gradient descent, CRF parameters may be trained.

IV. PROPOSED APPROACH/METHODOLOGY

In Diagram 3.1, a framework of the system is described including various methods and steps.

The first stage is to distinguish moving background elements from foreground items. For generating a pixel view of the foreground at each frame, we employ post-processing techniques and an adaptive background removal method. Then, in the foreground pixel map, we group related areas, and object properties like bounding box and centre of mass are computed [35].

Following background subtraction is tracking. Our video surveillance system employs an object level tracking algorithm. “We follow the item as a whole from frame to frame rather than tracking individual pieces of it, like human limbs” [24].

The identification of anomalous behavior (including carried or abandoned objects like bags) comes as the last phase. The procedure is as follows:

3.1 Background Subtraction

The proposed system employs a 3-phase procedure to determine foreground items from video sequence image [29]. The backdrop scene must be initialized initially. The literature uses a variety of modeling methods for the backdrop scene. We evaluated frame/temporal differencing, Gaussian mixture parameters by using OpenCV functions, and derived Gaussian mixture fraemwork of our system in order to assess the effectiveness and compare run-time efficiency of various background scene frameworks for object detection. The system's components that are involved in foreground detection are compared, and system’s Gaussian mixture framework is integrated with other modules to enable the flexibility of the etire detection system.

The background model and the most recent picture from the video are used to update the foreground object pixels in the backdrop subtraction method's next phase [27, 30]. This method depending on the operative background framework, which was updated to account for dynamic scene changes. The foreground pixel that was spotted has noise in it from the surroundings or the camera. Carry out the pixel-level post-processing techniques to eliminate the noise in the pixels of the foreground.

After obtaining the pixels of the foreground, the linked component approach is utilised to identify connected areas, and the subsequent step involves computing the object bounding rectangles. The labelled areas may not be connected as a result of flaws in the foreground segmentation method [33]. Therefore, merging such separate zones is necessary through experimentation in order to be successful. In the pixel-level post processing stage, several relatively tiny areas are also removed because of ambient noise. By applying the foreground pixel map, the area and the centre of mass of the areas corresponding to items are retrieved from the current video picture.

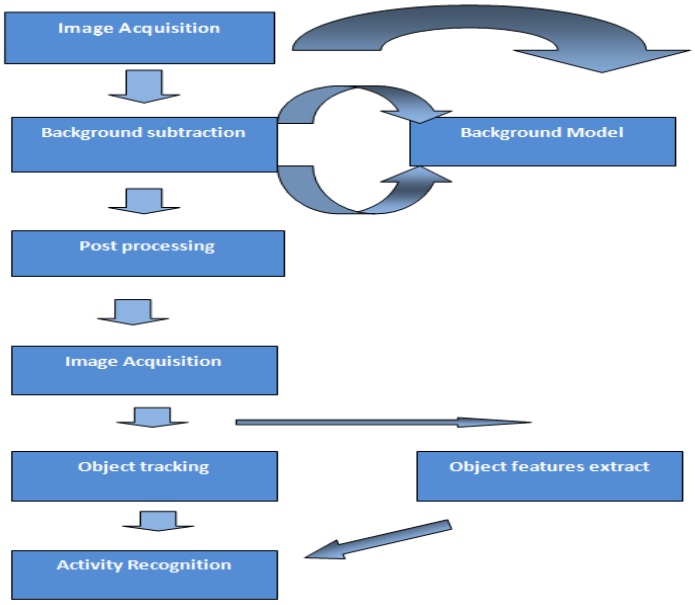


Diagram 3.1: Block Diagram of a System [23, 41]

3.1.1 Adaptive Gaussian Mixture Model

The values in GMM on viewing each pixel through time are modeled as a process of pixel, and the most recent individual pixel, X1,..., Xt, is made up of a combination of K Gaussian distributions. Equation (3.1) is used to calculate the likelihood of locating the current background pixel value [12].

$$P(\vec{X}_{\mathbf{j},t}|\vec{X}_{\mathbf{j},1},\ldots,\vec{X}_{\mathbf{j},t-1})=\sum^{K} \omega_{\mathbf{j},t}\times 5(\vec{X}_{\mathbf{j},t},\vec{\mu}_{\mathbf{j},t},\Sigma_{\mathbf{j},t}) \quad (3.1)$$

"Where K is the number of Gaussian distribution, ωj,t is an estimation of the weight of the jth Gaussian of the mixture at time, μj,t is the mean value and Σj,t is the corresponding covariance matrix and η is a Gaussian probability density function that is computed in equation (3.2) given in "[13].

$$\eta\left(\vec{X}_{j,t},\vec{\mu}_{j,t},\Sigma_{j,t}\right) = \frac{1}{2\pi^{n/2}\left|\Sigma_{j,t}\right|^{1/2}} e^{-\frac{1}{2}(\vec{X}_{j,t}-\vec{\mu}_{j,t})^{T}\Sigma_{j,t}^{-1}(\vec{X}_{j,t}-\vec{\mu}_{j,t})} \quad (3.2)$$

"Where n is the n-dimensional from vector X j,t. In this case, n = 3 because we adopt RGB color space and K depends on computational power and available memory, normally range is 3-5 [13].

Color is an important factor to describe objects. In order to find the probability distribution of color characteristics, we assume different color channels are independent from each other [11], so variation matrix is defined as in equation"(3.3) [13, 15].

$$\Sigma_{j,t} = \begin{matrix} (\sigma_{j,t}^2)^R & 0 & 0 \\ 0 & (\sigma_{j,t}^2)^G & 0 \\ 0 & 0 & (\sigma_{j,t}^2)^B \end{matrix} \quad (3.3)$$

Where $(\sigma_{j,t}^2)^R$, $(\sigma_{j,t}^2)^G$ and $(\sigma_{j,t}^2)^B$ are the RGB channel variances.

"Every time when a new pixel Xj,t is observed it is checked against the already existing K distributions. A match is defined as in equation 3.4 [15]."

$$|X_{j,t}^{x} - \mu_{j,t}^{x}| \leq 2.5 * \sigma_{j,t}^{x} \quad (3.4)$$

"Where x denotes R and B, respectively. If a match is found for some distribution, then eq. 6 is updated. If no distribution is matched among the existing K distributions then replace the least probability distribution with the new distribution using the mean, weight and variance of the current pixel $X_{j,t}$, the initial high variance and low weight, respectively [42]. The least probable distribution is finding out by the lowest ω/σ value. The prior weights of K distributions at time t, $\omega_{k,t}$ are updated" as the equation 3.5 [28].

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(S_{k,t}) \quad (3.5)$$

"Where α is the learning rate having the values between 0 to 1 and speed at which distribution parameters change depends on time constant $1/\alpha$.

$S_{k,t}$ is 1 if match is found and 0 for the remaining values. $\vec{\mu}_{j,t-1}$ and $\alpha_{j,t-1}$ are parameters for unmatched distributions that contain the same value and the parameters that match the new distribution are updated" using equation as below [26].

$$\vec{\mu}_{j,t} = (1 - \rho)\vec{\mu}_{j,t-1} + \rho X_{j,t}$$

$$\sigma_{j,t}^{2R} = (1 - \rho)\sigma_{j,t-1}^{2R} + \rho(R_{j,t} - \mu_{j,t-1}^{R})^2$$

$$\sigma_{j,t}^{2G} = (1 - \rho)\sigma_{j,t-1}^{2G} + \rho(G_{j,t} - \mu_{j,t-1}^{G})^2$$

$$\sigma_{j,t}^{2R} = (1 - \rho)\sigma_{j,t-1}^{2R} + \rho(R_{j,t} - \mu_{j,t-1}^{R})^2$$

$$\rho = \alpha * \eta(\vec{X}_{j,t}, \vec{\mu}_{j,t-1}, \sigma_{j,t-1})$$

Where, ρ is defined as the 2nd learning rate

"The Gaussian parameters must be adjusted when a match is found within the existent K Gaussian distributions [36]. The weights (ω) of all Gaussian distributions must be adjusted and the standard deviation (σ) and the mean (μ) are updated for the matched Gaussians, while unmatched Gaussians remain same" [15]. Update the weights, deviations and means using the equations (3.6) to (3.9) and ρ is calculated using equation (3.10).

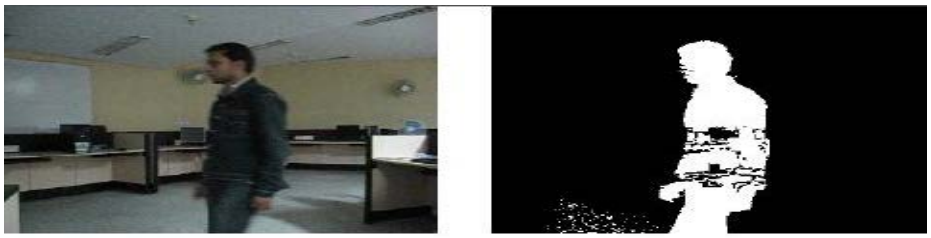
"After every updating operation, the K distribution are ordered by the value of ω/σ, and the most likely background distribution is always on the top of the K distribution then chose the first R distribution as the real background using equation (3.6).

$$R = arg_r \min(\Sigma^r \omega_k > T) \quad (3.6)$$

Where threshold T is the minimum fraction of background model or it is defined as the minimum prior probability of background to be in the image scene" [3].

"In order to get the faster adaptation of mean and the variance value, we just cut off the η component from the ρ definition. The purpose of updating the parameters in time is that σ will have a larger value than the proposed values in many literatures" [26]. "If any object moves suddenly than it will be detected using former learning rate while with the larger σ value the true background will get the dominant place. To make the background subtraction more efficient cut off the η value that save the time and space [13]. Then there is no requirement to store the value 5 ($\vec{X}j,t$, $\vec{\mu}_{j,t}$, $\Sigma_{j,t}$). Record the K distributions by the value $\omega_{k,t}$ instead of ω/σ, thus the computational load will be less. After this reduction the parameters that must be computed and stored are mean value vector $\mu = (\mu^R, \mu^G, \mu^B)$ and variance vector $\sigma = (\sigma^R, \sigma^G, \sigma^B)$ and weight $\omega_{k,t}$ of each model [12]. But three additional parameters ρ, η, ω/σ must be calculated and stored in original GMM. Therefore, computational load will be higher in original GMM. Thus performance of our method is more efficient than original GMM."

Diagram 3.2(a) depict the video footage, while the Diagram 3.2(b) displays the background-subtraction outcome.



(a) (b)

Diagram 3.2: Background Subtraction Results (a) Original picture (b) Picture after subtraction of background

3.1.2 Temporal Differencing

The simplest background removal methods use temporal frames differencing to detect moving objects. Additionally, because it is non-recursive, there is no need to save the history of frames in the buffer. As a result, this method uses less memory than other ones that are already in use. This technique subtracts the pixel values from the previous frames from the pixel values of the current frame [14]. The temporal differencing approach loses the halted item when a foreground object in a video clip stops moving because it is unable to recognize the difference between two consecutive frames. To detect halted items, then, specialized assisting algorithms were needed.

"We pre-set a temporal differencing algorithm that uses two consecutive frames. In the range [0, 255], let ln(k) represent the intensity value of the grey level at pixel location (k) and at time instance n of video frame sequence l". A moving pixel solves the equation (3.12) stated in [29] in a two frame temporal differencing approach.

$$|l_n(k) - l_{n-1}(k)| > r_n(k) \quad (3.12)$$

"Where, r_n is the pre-defined threshold. Hence, if any object has uniform colored regions then equation 4.14 fails to detect some pixels inside the region even if the object moves in the video [41]. The per pixel threshold, is initially set to a pre defined value and later updated as equation (4.13).

$$\tau_{n+1}(k) = \begin{cases} \alpha\tau_n + (1-\alpha)(\delta \times |l_n(k) - l_{n-1}(k)|), & k \in BG \\ \tau_n(k), & k \in FG \end{cases} \quad (3.13)$$

Where α, $\delta \in [0.0, 1.0]$ are the learning constants which determine the amount of information that is put to the background and threshold from the incoming image. Background image is a weighted temporal average of incoming picture sequences and threshold image is a weighted temporal average of times the difference between incoming image sequences and the background [6] if background pixels are thought of as time series."

3.1.3 Pixel Level Post Processing

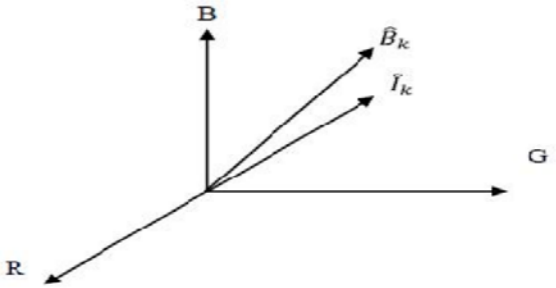
As in aforementioned paragraphs, the output of algorithms for foreground detection contains noise and post processing is necessary for making it suitable. The noise in foreground analysis is caused by a number of sources for example shadow and camera etc

3.1.3.1 SHADOW & NOISE REDUCTION

The difference between an item and its shadow must be made as the shadow causes problem in foreground recognition. The brightness parameter of the pixel of the shadow is lesser than the brightness of the corresponding pixels of the background, and the RGB colour vectors of the pixel in the shadow region are identical, with a slight variation, to the same colour vector of the corresponding pixels of the background [5]. "Allow I_k to represent the RGB colours of the current picture pixel at location and the colours of the corresponding background pixel in order to

$$\left(d_k = \frac{I_k}{\|I_k\|} \cdot \frac{B_k}{\|B_k\|}\right) < \tau$$

$$\|I_k\| < \|B_k\|$$



specify this.”

“Diagram 3.3: RGB vectors I_k (current image pixel) and B_k (background pixel)

Where τ is a predefined threshold that is close to 1. Dot product is used to check whether Ik and Bk. have same direction or not and if the dot product (dk) of normalized *I^k* and*B ^k* is close to 1, this implies that both vectors are in same direction with little amount of deviation [28]”.

The foreground pixel is subjected to the morphological techniques dilation and erosion in order to reduce noise. The method has been used to get rid of noisy foreground pixels that don't match the real foreground region and background pixels which are noisy that are inside or close to the actual foreground region [9]. Erosion removes one unit thick boundary pixels from foreground regions.

3.1.3.2 Connected Component determination

“The filtered foreground pixels are organised into connected components (blobs) by utilising a two level connected component algorithm” [4]. Calculate the bounding box of these regions after locating the specific blobs that stand in for the respective objects.

3.1.3.2 Post-Processing at Regions Level

After the pixel level noise has been eliminated, a few tiny areas still contain noise because of incorrect object segmentation. The average region size for each frame is determined in terms of pixels to remove this kind of noise. The foreground pixels map excludes regions whose size is less than a certain percentage of the average region size [41]. Due to segmentation mistakes, certain object pieces are discovered to be disconnected from the main body. Bounding boxes of areas that are close to one another are blended together to address this flaw. The outcome of the morphological operation and shadow eradication is displayed in Diagram 3.4.

Diagram3.4: Pixel-level post-processing output

3.1.3.3 Extracting Object Features

The characteristics of associated objects are extracted from the current scene after segmenting the foreground areas. These characteristics are the object's size (Si) and centre of mass (Ce). We just calculate the amount of pixels in order to

approximate the object's size. "Use the equation (3.16), found in [6], to determine the centre of mass Ce = (xC_e, yC_e) of an object *O* and to detect suspicious activity via video surveillance.

$$xC_i = \frac{\sum_i^k x_i}{k}, \quad yC_i = \frac{\sum_i^k y_i}{k} \tag{3.16}$$

Where *k* is the number of pixels in object *O*".

3.2 Tracking

Our video surveillance system employs an object level tracking algorithm. Instead of tracking individual elements of the item, like human limbs, we follow the thing from frame to frame. The information that was retrieved during the tracking phase is sufficient for the majority of video surveillance applications. Our method establishes a matching between objects from frame to frame by using the retrieved object characteristics, such as centre of mass, size, and bounding box [19, 39]. The tracking algorithm is capable of detecting carried and abandoned objects as well as objects occlusion and object identity distinction. Matching the new items in the current frame to the objects in the previous frame is the initial step in the object tracking algorithm.

The bi-partite graph *G(v,e)* stores the object matching. In this graph, the edges represent a match and the vertices represent the objects. The sizes of partition for prior items are denoted by by *v* in *G(v,e)* and new objects by *e*. Diagram 3.5 displays an example matching graph that utilizes the data from [16]'s observation. Iterate through a new object for each previous item O_a, and before doing so, determine whether the object O_b *(new)* in the new objects list is similar to O_a [38]. If two objects with centre of mass C_a and C_b are near to one another,

$$Distance\ (C_a, C_b) < \tau \tag{3.17}$$

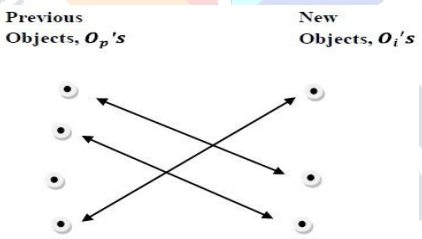


Diagram 3.5: Bi-partite matching graph [16]

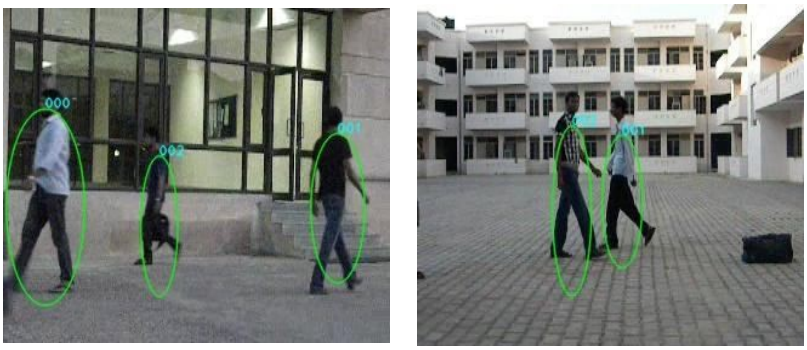
If two items are close to one another and within a threshold, a successful match between them need not always occur. We thus assess the similarity of two items in the next phase to enhance proper matching. The items' size ratio is used as a check to make sure that objects do not change size significantly between frames [21]. If the two things meet the conditions as mentioned below, they could be similar.

$$\frac{S_i}{S_j} < u \ \ or \ \ \frac{S_j}{S_i} < u$$

Where S_j is object size O_b and μ is a pre-defined threshold

The situation where a preceding item might match to more than one object would arise if the first two stages are carried out. Check the object O_a again after the second step to see whether there is already a match [24]. If object O_a does not already have a match, then connect the relevant vertices in the bipartite graph *G(v,e)* and go on to object O_b. If object O_a already has a match, then resolve the correspondence conflict by doing extra steps.

We try to determine whether of O_b or O_c is the proper match with the object O_a by comparing the correspondence of O_b and O_a with the correspondence of O_c and O_a. Utilising the separation between the centres of mass of objects O_a and O_b or O_c, correspondences between the objects are compared. Let *dsb* represent the separation between O_a*'s* and O_b*'s* centres of mass and *disc* represent the space between O_a*'s* and O_c*'s* centres of masses. If *disc*> *dsb*, the correspondence is resolved in favour of; if not, it favours O_b [38]. Refer diagram 3.6 for the ouput



(1) (2)

Diagram 3.6: Tracking results

3.3 Behavior or Activity Recognition

Application areas for activity detection include sports, where the activity may be described a hitting a ball. It may also involve a surveillance-related application, such as bag abandonment, bag carrying, or accident occurrence using traffic videos [1, 5]. Each frame's output from the tracking framework and extracting features serve as the input for the leftout or carried item identification method, which decides whether an object is left or holding or carried [36,20]. The outcome of the tracking phase includes the objects with their identity and the results of the features extraction phase. The detection procedure includes the following phases to address the issue of carried or dumped bags:

The first step is to identify the bag item.

The second step is to determine who they are.

The third step is to perform a test to look for unexpected behaviour.

Phase 1:

In order to identify the bag, we take the common of the area in all frame where the tracker region (X_l) and the foreground segmentation (Fs) results create an item blob. According to the likelihoods established in equations (3.19) and (3.20), a tiny, motionless blob can be assumed to be a bag component [5].

$$p_h(B^i = 1|\overline{X}^i_{1:l}) \propto N(S^i_l, \mu_h, \sigma_h) \quad (3.19)$$

$$p_k(B^i = 1|\overline{X}^i_{1:l}) \propto \exp(-\lambda v^i_l) \quad (3.20)$$

"Where S^i_l is the size of blob i at time l, p_h is the size likelihood, p_k is the velocity likelihood, B^i=1 indicates that blob i is a bag, μ_h is the mean bag blob size, σ_h is the bag blob variance, v_i is the blob velocity, and is a hyper-parameter. When frame-by-frame likelihoods are added without adjusting for blob lifespan, the long-living blob is more likely to be the carried or abandoned bag location [37]. Equation (3.21) combines p_h and p_k to determine the overall chance that a blob is a carried or abandoned bag item" [25].

$$p(B^i = 1|\overline{X}^i_{1:l}) \propto \sum_{t=i:T} N(S^i_l, \mu_h, \sigma_h)\, exp\,(-\lambda v^i_l) \quad (3.21)$$

The bag likelihood term $p(B^i = 1|\overline{X}^i_{1:l})$ gives preference to long lasting and small objects. The person is selected by thresholding the likelihood using equation (3.22) given in [25].

$$p\left(B^i = 1\middle|\overline{X}^i_{1:l}\right) > T_B \quad (3.22)$$

A shape template $\mathcal{T}^i$ is constructed from the longest frame segment below a low threshold v_t, to model what bag looks like when it is stationary [34]. Bag existence likelihood is determined for blob person by extracting image features from the binary image at stationary bag $\mathcal{L}_l$ and performing an element wise multiplication using equation (3.23) [5, 25].

$$p(E_l = 1/B^i) \propto \sum_c \sum_d \mathcal{T}^i(c,d) \times \mathcal{L}_l(c,d) \quad (3.23)$$

Where $E_l = 1$ indicates that bag exists at time l, and c and d are pixel indices [5].

Phase 2:

Examine the tracker's previous record to see when the bag first comes into view, which was decided by the chance that the bag was present earlier. If the tracker disappears and dies but the bag stays put, it must be the owner's tracker. If the tracker is still in the bag, we start looking for fresh tracker births in the area. The individual is considered to be the first nearby birth. If there is not any nearby birth is discovered, the possessor of the bag is missing and the next step is not necessary [36].

Phase 3: The final task, detecting whether the bag is dumped or carried, is simple after the bag and the person have been spotted and their locations are known [5]. Unusual activity is identified if the interspace between the centre of mass of the carried or dumped object and the person exceeds a threshold value and keeps increasing with the time.

V. EXPERIMENTAL RESULTS AND EVALUATION

The framework of the system is able to take all types of video sequences. The programme of the system is written C++ using the OpenCV library in Linux OS.

If a moving or static item enters the scene and then pauses, it will blend into the backdrop. However, the blending into the background is not desired because doing so would make the item disappear or lost. In order to prevent foreground and background from merging, this application needs a temporal control technique. We can observe that the GMM updating process is the cause of the object's merging.

In cases when occlusion and shadow are eliminated, our results for activity detection are more stable and trustworthy. For the detection of suspicious or unusual activity, the norm is to check if the interspace between the point of the centre of mass of the carried or dumped object and the man is more than the threshold point and is increasing.

4.1 Comparison of the Results of Background Subtraction

A person is shown carrying a bag in Diagram 3(a), and the background removal result is displayed in Diagram 3 (b). Diagram 3 (c) depicts a person leaving their bag behind as they walk away from it, whereas Diagram 3(d) shows simply a person being seen and no stationary bags being seen.

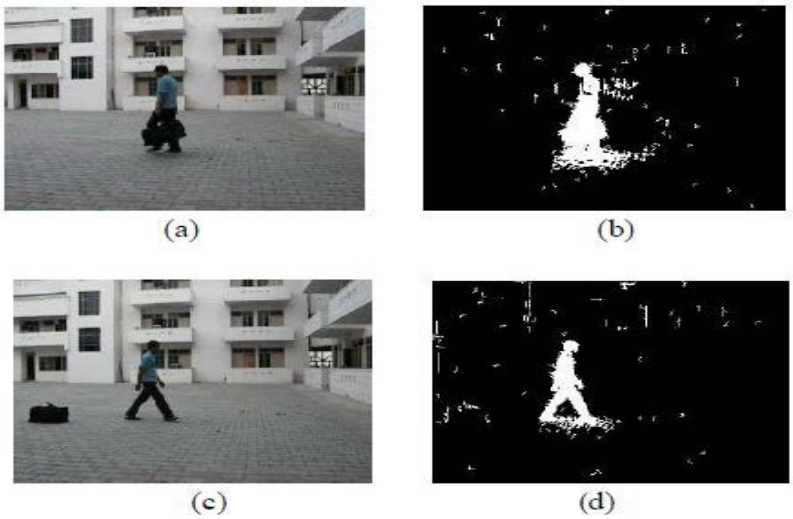


Diagram 3 : GMM results are shown in Open CV.

Background tracking and subtraction results

A person is shown carrying a bag in Diagram 4(a), and the background subtraction result is displayed in Diagram 4(b). Diagram 4(c) depicts a man walking away from a bag they had abandoned, whereas Diagram 4(d) shows the detection of both the person and the bag.

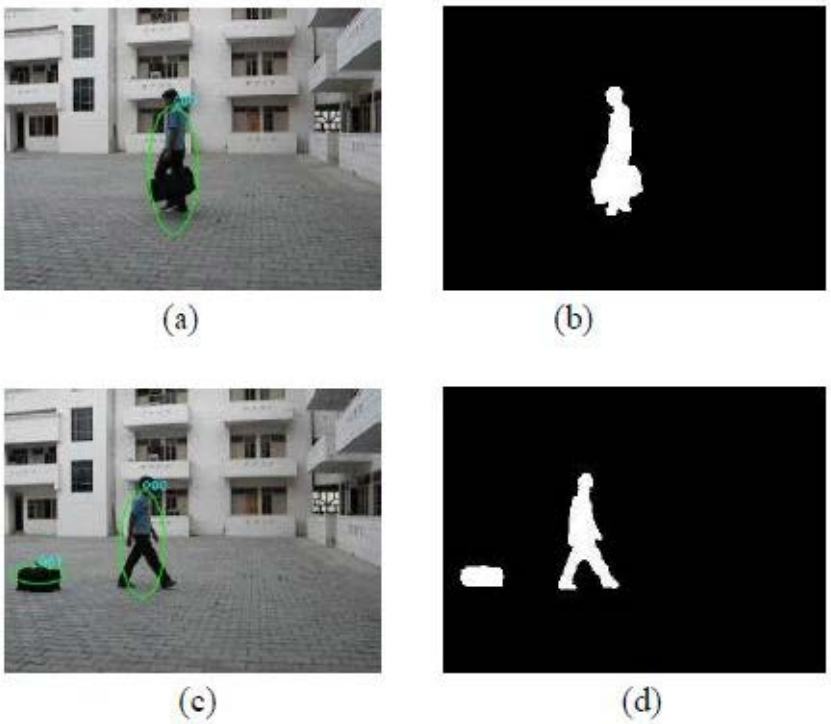


Diagram 4: Results of our background removal and tracking

4.2 Results of Activity Recognition

The suggested technique is one and a half times quicker than the primary Gaussian mixture framework, and the proposed system results for activity detremination are more stable and accurate when occlusion and shadow removal are taken into account. The discovery of suspicious or unusual activity is confirmed, if the interspace between the point of the centre of mass of the carried or dumped object and the man is more than the threshold point and is increasing.

4.3 Abandoned bag detection results:

OUTPUT OF VIDEO CLIP 1:

A person is walking in Diagram 5(a), and Diagram 5(d) displays the background subtraction result in relation to it. Diagram 5(e) is showing the background elimination output for the bag which was dumped in Diagram 5(b) by the man. Diagram 5(c) depicts a person leaving an abandoned bag. As a consequence, anomalous activity is recognised, and Diagram 5(f) displays the background subtraction result.

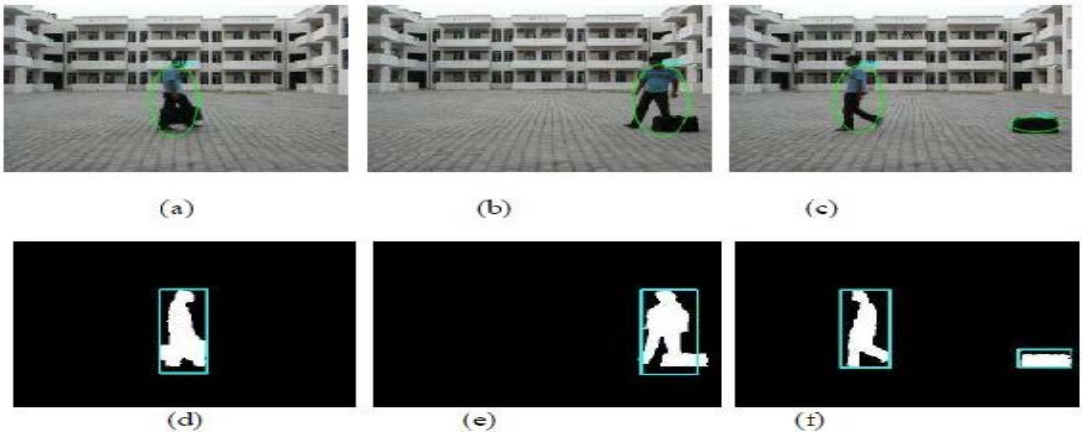


Diagram 5: The detection of abandoned bags is shown in Video 1.

Output of Video clip 2:

A person is walking in Diagram 6(a), and Diagram 6(d) displays the background subtraction result that corresponds to it. Diagram 6(e) displays the background elimination result for the bag that was dumped in Diagram 6(b) by the man. Diagram 6(c) depicts a

person leaving an abandoned bag. As a consequence, anomalous activity is recognised, and Diagram 6(f) displays the background subtraction result.

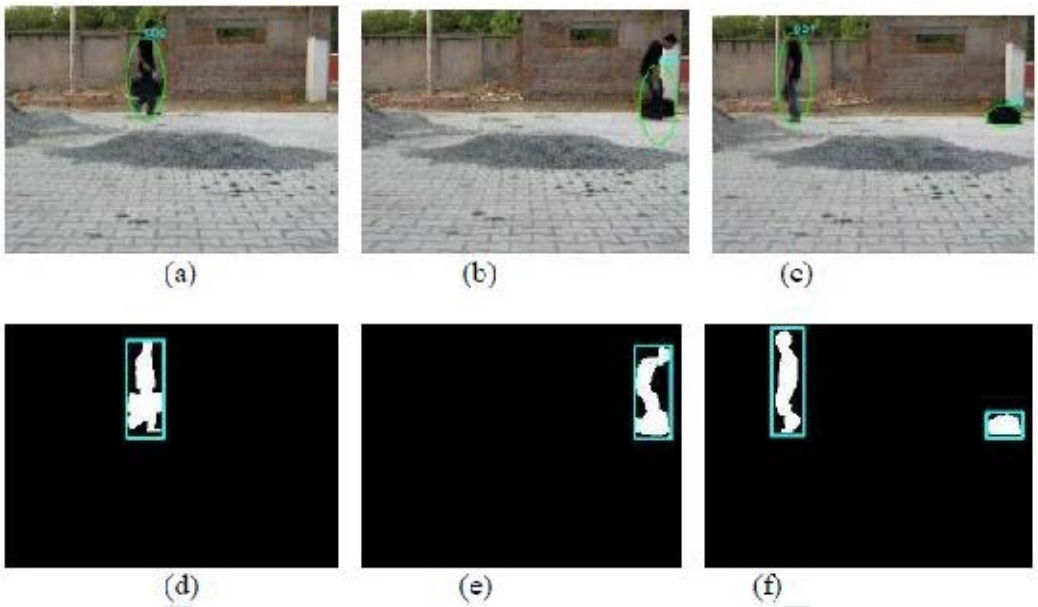


Diagram 6: Results of abandoned bag identification in video 2

Output of Video clip 3:

A person is walking in Diagram 7(a), and Diagram 7(d) displays the background subtraction result that corresponds to it. Diagram 4.5(e) displays the background subtraction result for the bag that the individual in Diagram 7(b) abandoned. Diagram 7(c) depicts a person leaving an abandoned bag. As a consequence, anomalous activity is recognized, and Diagram 7(f) displays the background subtraction result.

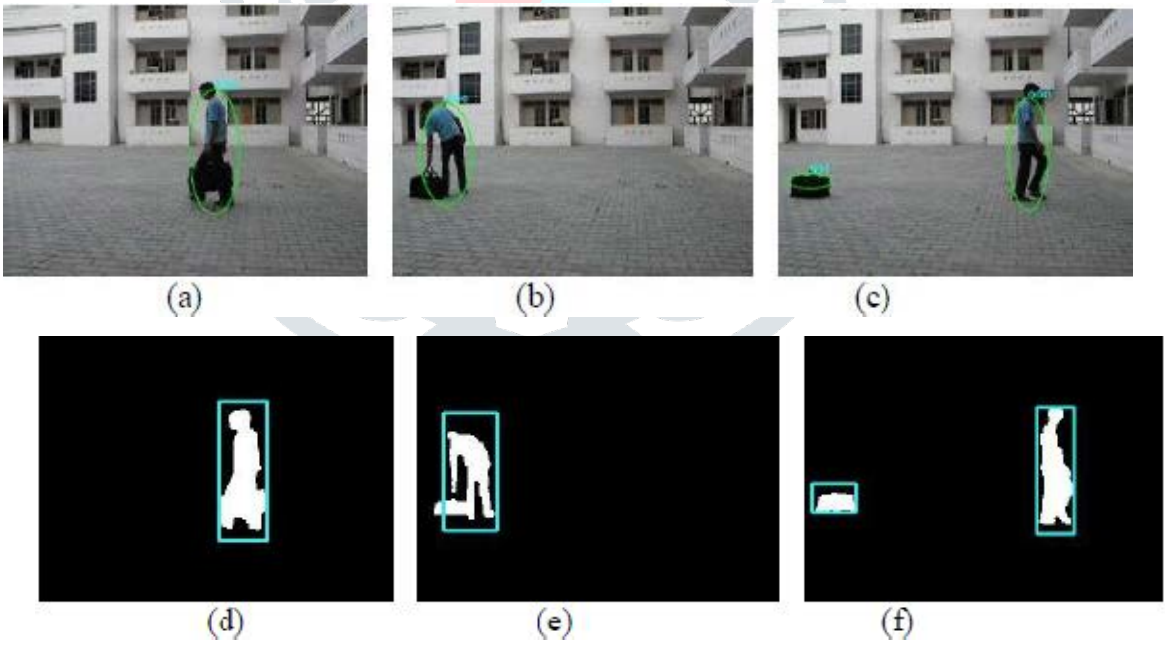


Diagram 7: Results of abandoned bag identification in video 3

4.4 Results of the detected carried object*:*

Output of Video clip 4:

A person is walking in Diagram 8(a), and Diagram 8(d) displays the background subtraction result corresponding to it. Since the bag in Diagram 8(a) is a background item, it is not recognized as an object in foreground in the Diagram 8(d). A person is shown carrying a bag in Diagram 8(b), and the resulting backdrop subtraction is displayed in Diagram 8(e). Diagram 8(f) shows the background subtraction result when the individual in Diagram 8(c) moves away from the bag position, which causes unusual activity to be noticed. Diagram 8(f) demonstrates that the bag has been relocated from its original place since the bag location is identified as a foreground item.

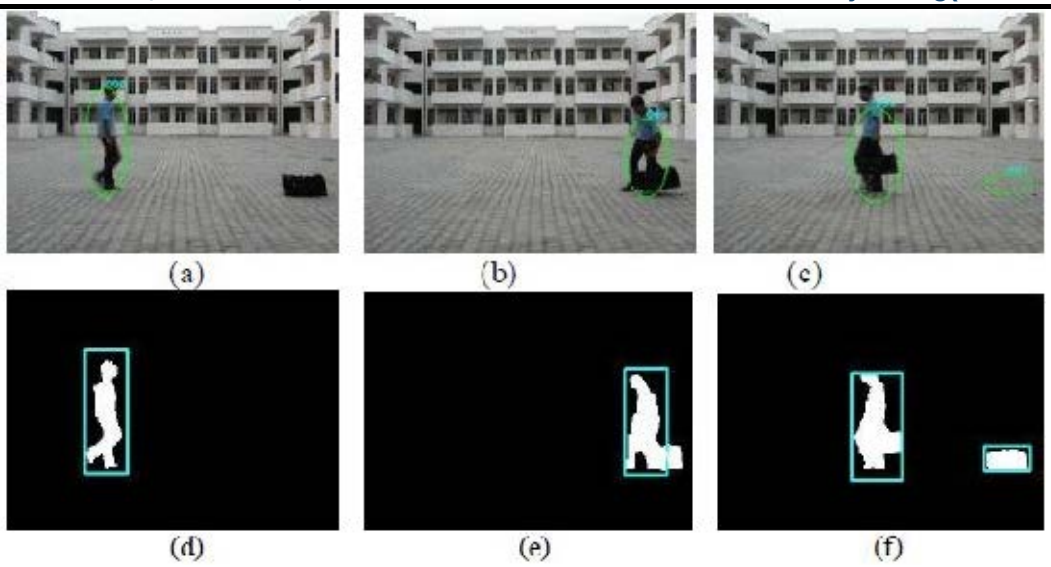


Diagram 8: Results of carried bag identification in video 4

Output of Video clip 5:

A person is walking in Diagram 9(a), and Diagram 9(d) displays the background subtraction result that corresponds to it. Since the bag in Diagram 9(a) is a background item, it is not recognized as a foreground object in Diagram 9(d). Diagram 9(e) displays the background subtraction result for the individual standing close to the bag in Diagram 9(b). Diagram 9(c) depicts a person carrying a bag as they leave the bag's position. This results in the detection of unusual activity, and Diagram 9(f) displays the background subtraction ouput for bag location and man both spotted by the application. The bag is shown as a foreground item in Diagram 9(f), indicating that it has been relocated from its original position.

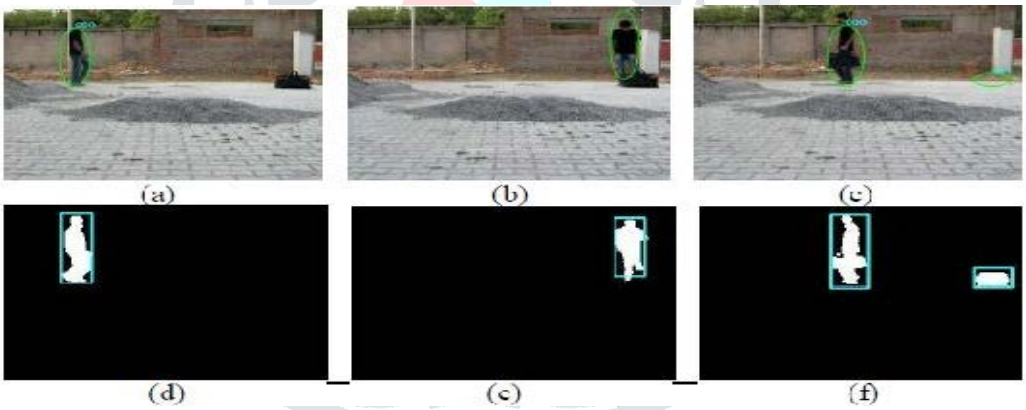


Diagram 9: Results of carried bag identification in video 5

Output of Video clip 6:

A person is walking in Diagram 10 (a), and Diagram 10 (d) displays the background subtraction result that corresponds to it. Since the bag in Diagram 10(a) is a background item, it is not recognized as a foreground object in Diagram 10(d). Diagram 10(b) depicts a man close to a bag, while Diagram 10(e) displays the outcome of removing the backdrop. Diagram 10(c) depicts a person carrying a bag and leaving the bag's position. As a consequence, unusual activity is recognized, and Diagram 10(f) displays the background subtraction output for man with the bag and specific place of the item spotted. The fact that the bag is identified as a foreground article in Diagram 10(f) indicates that it has been relocated from its original position.

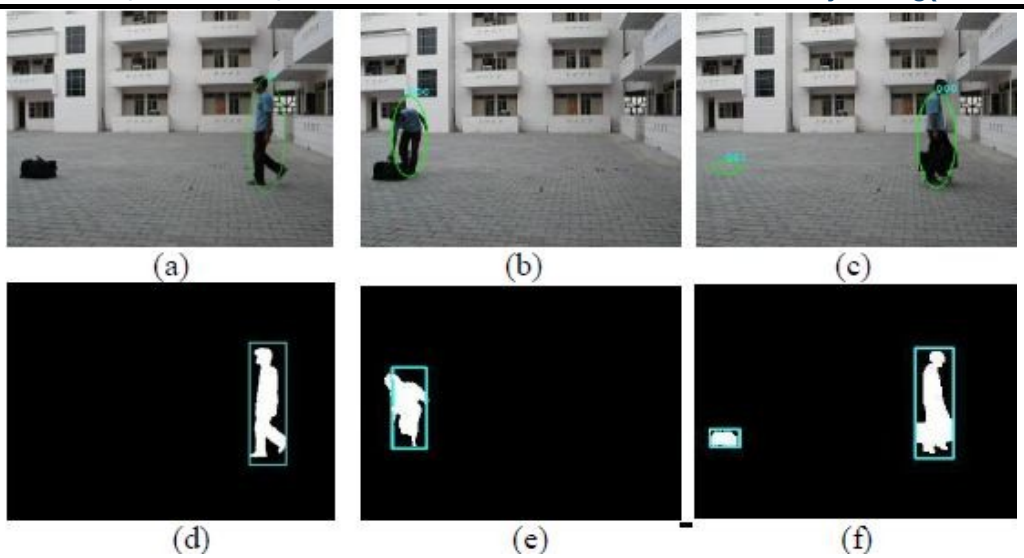


Diagram 10: Results of carrying bag identification in Video 6

Table-1: The suggested technique clearly illustrates its superiority over surveillance systems employing GMM provided by OpenCV for activity detection.

TABLE-1. GMM AND PROPOSED METHOD COMPARISON

Sample videos (SV)	**Length of Video sequence**	**Count of Frames in the Video clip**	**Time in Second for the suggested technique utilising Improved GMM**	**Time for the system utilising OpenCV (sec) and GMM**
SV 1	14	321	30	36
SV 2	15	348	35	41
SV 3	10	252	20	22
SV 4	11	251	19	23
SV 5	10	249	19	24
SV 6	13	247	25	31

The suggested solution has been tried out on several videos in various scenarios. "Different settings were chosen in order to highlight the proposed method's dependability and robustness. We defined the detection rate (DR) and false alarm rate (FAR) in order to provide a quantitative assessment of error [26].

$$DR = TP_{+VE}/(TP_{+VE} + FN_{-VE}) \quad (4.1)$$

$$FAR = FP_{+VE}/(TP + FP_{+VE}) \quad (4.2)$$

Where Actual detected foreground regions = TP_{+VE} (true positives); Detected regions that do not correspond to actual foreground region= FP_{+VE} (false positives); and moving objects that do not detected= FN_{-VE} (false negatives)"

Diagram 11 graph demonstrates that the suggested technique is more reliable than both the conventional GMM and the Temporal Differencing method. Traditional GMM and temporal differencing both experience an increase in false negative as background

complexity rises. Both of the current approaches identify more areas than the suggested method does that are not true foreground items.

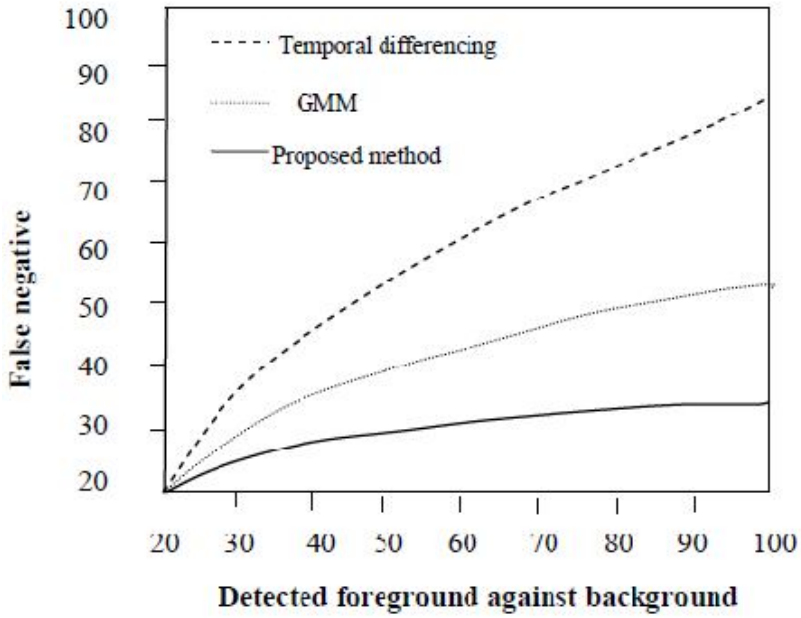


Diagram 11

VI. CONCLUSION

In conclusion, this thesis has presented a comprehensive investigation into the domain of Suspicious Activity Recognition in Video Sequences (SARVS). In this study, we discussed a number of video surveillance application strategies. We put various object identification algorithms into practise and contrasted the outcomes. Regarding the accuracy of object recognition and computational complexity, the adaptive GMM background removal approach yields the most encouraging findings.

The suggested object tracking method follows whole-body objects in a series of frames with success. Our studies on example applications demonstrate that matching based on correspondence strategy produces positive results and don't require complex strategies for tracking a large body of items.

Histogram-based matching technique determines the identities of the items entering into an occlusion following a split while managing basic object occlusions. A pixel-based technique, however, is more practical for handling object occlusions in cluttered environments. The proposed system can be upgraded with further study and research to work for streaming videos and real time tracking and detection of suspicious activity for enhanced security and surveillance.

While the presented research achieved notable advancements in Suspicious Activity Recognition in Video Sequences, there remain several avenues for future exploration and improvement in this field. Some potential areas of future research include:

1. Incremental Learning: Investigating incremental learning techniques to continuously update the SARVS system's knowledge and adapt to new and emerging suspicious behaviors, enhancing its ability to cope with evolving security threats.
2. Multi-modal Data Fusion: Integrating data from various sources, such as audio, thermal imaging, and social media, to enrich the SARVS system's perception and improve its accuracy in recognizing suspicious activities under different environmental conditions.
3. Real-time Deployment: Optimizing the SARVS system for real-time deployment on resource-constrained devices, enabling swift and efficient processing of video streams in live surveillance scenarios.
4. Anomaly Detection: Exploring unsupervised anomaly detection techniques to complement the supervised SARVS system, allowing it to identify previously unseen and novel suspicious activities without explicit training data.
5. Human-in-the-loop Systems: Developing human-in-the-loop SARVS systems, where human operators collaborate with the automated system to enhance decision-making and reduce false positives.
6. Privacy Considerations: Addressing privacy concerns related to video surveillance by developing privacy-preserving SARVS techniques that anonymize sensitive information while still ensuring effective activity recognition.

VII. FUTURE WORK

We present a surveillance system that works on offline videos so it is required to convert it into real time. No background subtraction algorithm is perfect for true object detection, so our method needs improvements in handling partially object occlusions, sudden illumination changes, and darker shadows. To enhance object detection results and eliminate inaccurate object segmentation, higher level semantic analysis extraction steps would be used. Other possible avenues for future work include using multiple cameras views that can reduce the object occlusion problem and investigating methods for maintaining object identities in the tracker better [12]. Usually real world scenarios are more complicated than the scenarios we presented here, in terms of number of persons involved in the activities and variation in execution style. So more sophisticated algorithms are needed to consider to handle such complexities. This system can be used as an initial base system for advanced research in the field of video surveillance system.

VIII. REFERENCES

1. Chi-Hung Chuang, Jun-Wei Hsieh, Luo-Wei Tsai, Sin-Yu Chen, and Kuo-Chin Fan, "Carried Object Detection Using Ratio Histogram and its Application to Suspicious Event Analysis," *IEEE Transactions on Circuits and Systems for VideoTechnology,*vol.19,no.6,pp.911–916,June2009.
2. Gian Luca Foresti, Lucio Marcenaro, and CarloS. Regazzoni, "Automatic Detection and Indexing of Video Event Shots for Surveillance Applications, "*IEEE Transactionson Multimedia*, vol.4,no.4,pp.459–471,2002.
3. Javier Varona, Jordi Gonzàlez, Ignasi Rius, and Juan José Villanueva, "Importance of Detection for Video Surveillance Applications," *Optical Engineering*, vol. 47, no. 8,August2008.
4. Jie Yang, Jian Cheng, andHanqing Lu, "Human Activity Recognition Based on the Blob Features," in *Proc. IEEE International Conference on Multimedia and Expo*, pp.358 – 361,June2009.
5. Fengjun Lv, Xuefeng Song, BoWu, Vivek Kumar Singh, andRamakant Nevatia,"Left-Luggage Detection using Bayesian Inference," in *Proc. IEEE 9th InternationalWorkshoponPerformanceEvaluationofTrackingandSurveillance*,NewYork,USA,pages 83–90,June2006.
6. Ahmed FawziOtoom, Hatice Gunes, and Massimo Piccardi,"Automatic Classification of Abandoned Objects for Surveillance of Public Premise," in *Proc.IEEE Congresson Image and Signal Processing*,vol.4,pp.542–549,May2008.
7. Pavan Turaga, Rama Chellappa, V. S. Subrahmanian, and Octavian Udrea, "Machine Recognition of Human Activities:A Survey,"*IEEE Transactionson Circuits and Systems for Video Technology*,vol.18,no.11,pp.1473–1488,Nov.2008.
8. Fadhlan Hafiz, A.A. Shafie, M.H. Ali, and Othman Khalifa, "Event-Handling BasedSmart Video Surveillance System," *International Journal of Image Processing (IJIP)*,vol.4,no.1,2008.
9. Feng Niu and M.Abdul-Mottaleb,"View Invariant Human Activity Recognition Basedon

Shape and Motion Features," in*Proc.IEEE Sixth International Symposiumon Multimedia Software Engineering*, pp.546–556,2004.

10. Md.Zia Uddin, J.J.Lee, and T.S.Kim, "Independent Component Feature-based Human Activity Recognition via Linear Discriminant Analysis and Hidden MarkovModel," *inProc.IEEE 30th Annual International Conferenceon Engineering in Medicine and Biology Society*,pp.5168–5171,Aug.20–25,2008.

11. G.Lavee, E.Rivlin and M.Rudzsky, "Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications andReviews*,vol.39,no.5,pp.489–504,Sept.2009.

12. T. Bouwmans, F. El Baf, and B. Vachon, "Background Modeling using Mixture ofGaussiansforForegroundDetection-ASurvey,"*RecentPatentsonComputerScience*,vol. 1,no.3, pp.219-237, Nov.2008.

13. ZhenTangandZhenjiangMiao,"FastBackgroundSubtractionUsingImprovedGMM and Graph Cut," in *Proc. IEEE International Conference on Image and SignalProcessing*,vol.4, pp.181–185,2008.

14. J.MikeMcHugh, JanuszKonrad, Venkatesh Saligrama, and Pierre-MarcJodoin, "Foreground-Adaptive Background Subtraction," *IEEE Signal Processing Letters*,vol.16,no.5,pp. 390–393,May2009.

15. Li Ying-hong, Xiong Chang-zhen, Yin Yi-xin, Liu Ya-li, "Moving Object Detection Basedon Edged Mixture Gaussian Models," in *Proc. IEEE International Conferenceon Intelligent Systems and Applications*,pp.1-5,2009.

16. Aishy Amer,"Voting-Based Simultaneous Tracking of Multiple Video Objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp.1448-1462,2005.

17. Junzo Watada, Zalili Binti, andMusaand Graduate, "Tracking Human MotionsforSecurity System," in *Proc. IEEE SICE Annual Conference*, pp. 3344 – 3349, August2008.

18. Nizar Zarka, Ziad Alhalah, and Rada Deeb, "Real-Time Human Motion Detection andTracking," in *Proc. IEEE International Conference on Information and Communication Technologies:from Theory to Applications*, pp.1-6,2008.

19. Robert Bodor, BennettJackson, and Nikolaos Papanikolopoulos,"Vision-Based Human Tracking and Activity Recognition,"in*Proc.of 11th Mediterranean Conference on Control and Automation*,vol.1,pp.18-22,2003.

20. Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,vol.25,no.10, pp.1337–1342,2003.

21. Tao Gao, Zheng-guang Liu, Wen-chun Gao, and Jun Zhang "Robust Tracking andObject Classification towards Automated Video Surveillance," in *Proc. International Conference on Image Analysis and Recognition*, pp.463–470,2004.

22. Liang Wang, Weiming Hu, and Tieniu Tan, "Recent Developments in Human MotionAnalysis,"*PatternRecognition*,vol.36,no.3,pp.585–601,2003.

23. Teddy Ko"ASurvey on Behavior Analysis in Video Surveillance for Home and SecurityApplications,"in*Proc.IEEE37thInternational Conferences on Applied Imagery Pattern Recognition*, pp. 1–8, 2008.

24. Mayssaa Al Najjar, Soumik Ghosh, and Magdy Bayoumi, "Robust Object TrackingUsing Corresponding Voting for Smart Surveillance Visual Sensing Nodes," in *Proc.IEEE Sixteenth International Conference on Image Processing*,pp.1133–1136,2009.

25. Kevin Smith, Pedro Quelhas, and Daniel Gatica-Perez, "Detecting Abandoned Luggage Items in a Public Space,"in *Proc.9th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, NewYork, USA, pages 75 –82,June2006.

26. H.L. Ribeiro and A. Gonzaga, "Hand Image Segmentation in Video Sequence by GMM:A Comparative Analysis," in *Proc. IEEE 19th Brazilian Symposium on Computer Graphics and Image Processing*, pp.357–364,2006.

27. Qin Wan and Yaonan Wang,"Background Subtraction Based on Adaptive Non-parametric Model, " in *Proc IEEE 7th World Congresson Intelligent Control and Automation*, pp. 5960–5965, June2008.

28. Shenz Zun-bing and Cui Xian-yu,"An Adaptive Learning Rate GMMf or Background Extraction," *Optoelectronics Letters*,vol.4,no.6,pp. 460-463,Nov.2008.

29. Julien Pilet, Christoph Strecha, and PascalFua,"Making Background Subtraction Robust to Sudden Illumination Changes," in *Proc. European Conferenceon Computer Vision, Marseille, France*, pp.567-580,Oct.2008.

30. Neil Robertson, Ian Reid, and Michael Brady, "Automatic Human Behavior Recognition and Explanation for CCTV Video Surveillance," *Security Journal*, vol.21,pp. 173-188,July2008.

31. By Sven Fleck and Wolfgang Straqer, "Smart Camera Based Monitoring System and its Application to Assisted Living," *Proceedings of IEEE*, vol. 96, no. 10, pp. 1698 –1714,2008.

32. Wanqing Li, Igor Kharitonenko, Serge Lichman, and Chaminda Weerasinghe,"A Prototype of Autonomous Intelligent Surveillance Cameras," in*Proc.IEEEInternational Conferenceon Video and Signal Based Surveillance*, pp.101-107,2006.

33. LIY ingjieand YINY ixin, "Towards Suspicious Behavior Discovery in Video Surveillance System,"in*Proc.IEEE 2nd International Workshop on Knowledge Discovery and Data Mining*,pp. 539–541,2009.

34. Dima Damen and David Hogg,"Detecting Carried Objects in Short Video Sequences," in *Proc. of the 10th European Conference on Computer Vision: Part III*, Marseille,France,pp. 154–167,2008.

35. WeiyaoLin, Ming-TingSun, Radha Poovandran, andZhengyou Zhang,"Human Activity Recognition for Video Surveillance,"in*Proc.IEEE International Symposium on Circuits and*

Systems, pp.2737–2740,May2008.

36. Meghna Singh,Anup Basu,andMrinal Kr.Mandal, "Human Activity Recognition Based on Silhouette Directionality," *IEEE Transactions on Circuits and Systems forVideoTechnology*,Vol.18,No.9,pp.1280–1292,Sept.2008.

37. Thanarat Horprasert, David Harwood, and Larry S.Davis,"A Robust Background Subtraction and Shadow Detection," in *Proc. Asian Conference on Computer Vision*,Taipei,Taiwan,pp.983–988,2000.

38. Jiang Danand Yu Yuan,"A Multi object Motion Tracking Method for Video Surveillance,"in*Proc.IEEE 8th A CIS International Conferenceon Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*,vol.1,pp. 402 –405,2007.

39. Ying Fang, Huiyuan Wang, Shuang Mao, and Xiaojuan Wu, "Multi-Object Tracking Based on Region Corresponding and Improved Color-Histogram Matching," in *Proc. IEEE International Symposiumon Signal Processing and InformationTechnology*,pp.1–4,2007.Chiraz Ben Abdelkader and Larry Davis, "Detection of People Carrying Objects: AMotion-BasedRecognition Approach," in *Proc. IEEE 5th International Conferenceon Automatic Face and Gesture Recognition*,pp.378–383,2002.

40. Kenneth Ellingsen, "Salient Event Detection in Video Surveillance Scenarios," in *Proc. ACM First workshop on Analysis and Retrieval of Events/Actions and Work flows in Video Frames*,pp. 57-64, 2008.

41. Hua Zhong, JianboShi, and Mirk´ o Visontai, "Detecting Unusual Activity in Video,"in*Proc.ofIEEEComputerSocietyConferenceonComputerVisionandPatternRecognition*,vol. 2, pp.819-826,2004.

42. Gary Bradski and Adrian Kaehler, *Learning Open CV*, First Edition, Sebastopol: O'Reilly Media, 2008.