2013

Bayesian inference application to burglary detection

Ishan Singh Bhale
Louisiana State University and Agricultural and Mechanical College, ishanbhale@hotmail.com

BAYESIAN INFERENCE APPLICATION TO BURGLARY DETECTION

A Thesis

Submitted to the Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirement for the degree of
Master of Science in System Science

in

The Computer Science and Engineering
Division of
The School of Electrical Engineering and Computer Science

by
Ishan Singh Bhale
B.E., Information Technology, RGPV, India, 2010
May 2013

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. Supratik Mukhopadhyay for supporting and guiding me throughout this thesis, and giving me the opportunity to work on the surveillance project. His guidance , advise, discussions were invaluable to me. I am grateful to Dr. Konstantin Busch, Dr. Jianhua Chen, and Dr. Nathan E. Brener to be part of my thesis committee and sparing their precious time.

I am grateful to computer science department of LSU. I would also like to thank the computer vision research group at LSU for all the discussions, cooperation, and help. The activity detection module that was important part of this whole complex activity recognition system was developed by the computer vision research group under the guidance of Dr. Supratik Mukhopadhyay.

I am thankful to Saikat Basu, Malcolm stagg, Robert DiBiano, Manohar Karki, for all the cooperation. I am also thankful to my family, friends who supported me, and kept me motivated all through my coursework and research work.

TABLE OF CONTENTS

Abstract

Real time motion tracking is very important for video analytics. But very little research has been done in identifying the top-level plans behind the atomic activities evident in various surveillance footages [61]. Surveillance videos can contain high level plans in the form of complex activities [61]. These complex activities are usually a combination of various articulated activities like breaking windshield, digging, and non-articulated activities like walking, running. We have developed a Bayesian framework for recognizing complex activities like burglary. This framework (belief network) is based on an expectation propagation algorithm [8] for approximate Bayesian inference. We provide experimental results showing the application of our framework for automatically detecting burglary from surveillance videos in real time.

Chapter 1

Introduction

1.1 Motivations

Video surveillance systems have become increasingly important for national security. Object tracking and motion classification are two important characteristics for any such surveillance system. However, once an object is tracked and its motion has been classified into a standard category by comparing it against a database of actions, the hard part is to use these actions or sequence of actions to discover activities that are unusual or seek attention [61]. Currently this is done by human operators who watch the output of surveillance cameras continually for unusual activities. However, for hours and hours of video data, this becomes a Herculean task and hence calls for an automated system that could track the objects, classify the motion, and reason about the top level plans of the subjects in the videos. Although many trackers [34,35,36,37,38] and motion classifiers [40,41,42,43] are available in industry today, none of them have the ability to reason about top level plans involving complex activities like robbery, burglary, or escapade.

1.2 Our Contributions

Real time motion tracking is very important for video analytics. But very little research has been done in identifying the top-level plans behind the atomic activities evident in various surveillance footages. Surveillance videos can contain high level plans in the form of complex activities [61]. These complex activities are usually a combination of various articulated activities like breaking windshield, digging, and non-articulated activities like walking, running. We have developed a Bayesian framework for recognizing complex activities like burglary. This framework (belief network) is based on an expectation propagation algorithm [8] for approximate Bayesian inference. We provide experimental results showing the application of our framework for automatically detecting burglary from surveillance videos in real time.

1.3 Framework

Figure 1.1 explains the architecture of our complex activity recognition system. The input video is initially stabilized to remove jitter and noise. The stabilized video is then fed to the object tracking module. The object tracking module tracks the moving objects in the video. The tracked video is then fed to the atomic activity recognition module to track articulated activities like digging, breaking windshield, and non-articulated activities like person walking, running, etc. An articulated activity is one that does not involve any translational motion; translational motion involves motion of only a part or parts of a body, whole body does not move. Such activities cannot be identified from a track (track stores path, velocity, color, and size of a moving object over time); they require analysis of the vertical and horizontal histograms (horizontal histograms and vertical histograms contain the mean of the intensity values of all the pixels along a particular direction) across frames [60]. The histogram contains the mean of the intensity values of all the pixels at a particular x or y coordinate; sometimes they require additional features like HOG (Histogram of Oriented Gradients) [59]. A non-articulated activity is one that involves translational motion; such an activity can be directly identified from a track, non-articulated activities are also called as track-based activities. The observed activities are then combined using the Bayesian framework based complex activity recognition module to infer complex activities like burglary. Complex activities are combination of many atomic activities.

1.4 Thesis Organization

Chapter 2 surveys existing work related to this thesis. Chapter 3 explains the system architecture for complex activity recognition.

Chapter 4 explains our framework for reasoning about complex activities. This chapter introduces the probabilistic complex activity recognition framework. This is followed by the description of the algorithm of this complex activity recognition, and its implementation.

Chapter 5 describes the application of our complex activity recognition framework to identifying complex activity like burglary in full motion video. We provide experimental results in this chapter in terms of a confusion matrix and a ROC Curve. Chapter 6 concludes the thesis and describes future work.

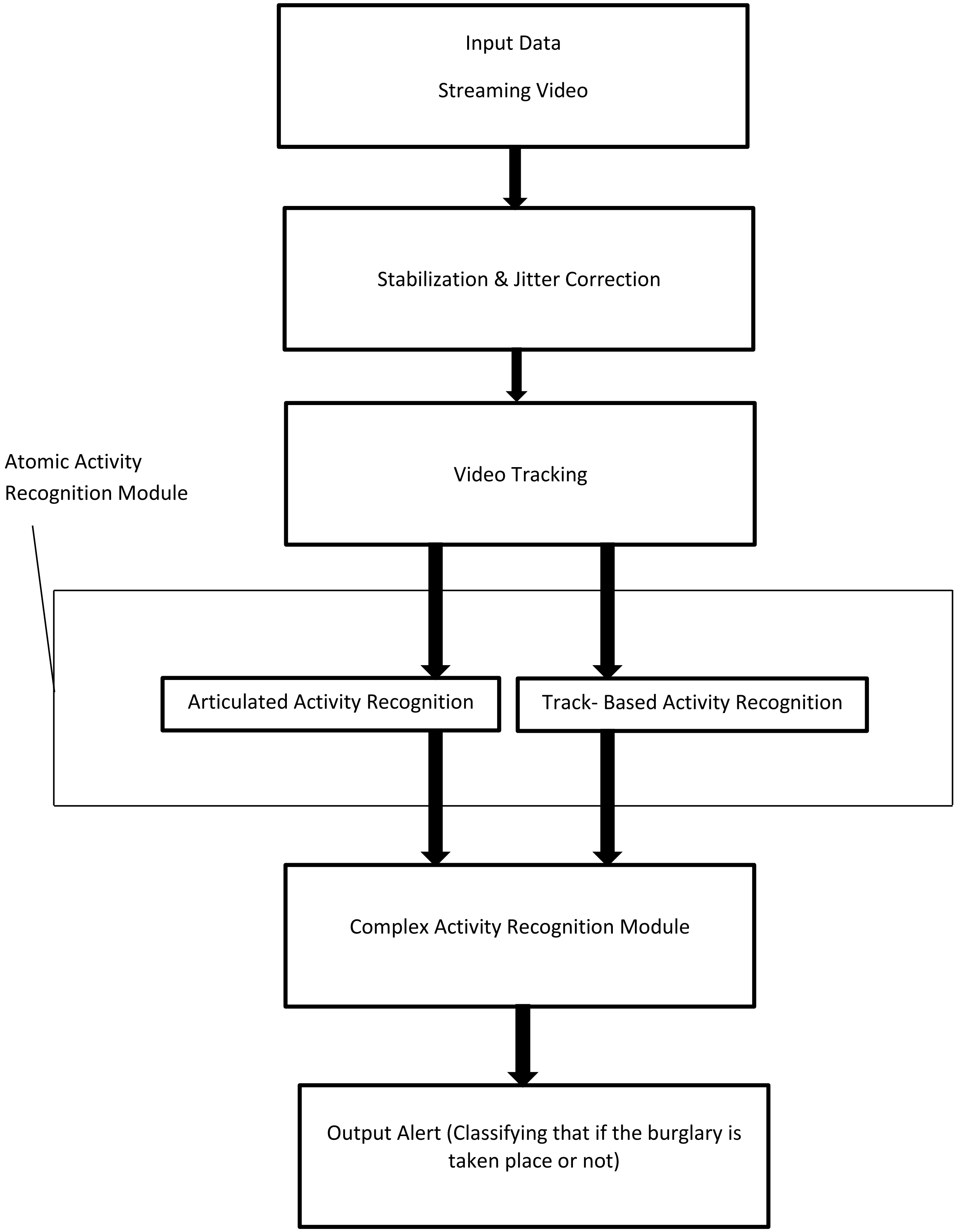


Figure 1.1: System Architecture [61]

Chapter 2

Related Work

Complex activity recognition from streaming videos has received a lot of attention in recent times. In [32], the authors have developed a system for recognizing activities involving teams of people, activities occurring within the same team, and interactions between a team and an individual, for e.g., stealing, fight, arrest, etc.. They have composed the individual actions of each person in the team to infer the composite activities performed by the whole team. They have used their system to identify various activities like cops taking someone into custody, a crowd engaging in violence, etc. The system in [32] can only reason about interactions between humans; while in this thesis we consider activities that involve interactions between vehicles and humans as well as interactions between humans.

In [33], the authors have surveyed the field of automated complex activity recognition from videos. According to them, a complex activity recognition system first takes the input video data and preprocesses it; this step involves correction of damaged frames; the video is then stabilized to correct jitter and noise in the video, and then fed to the next level for further processing; this step involves tracking the various moving bodies, detect various actions, etc.. Finally logical reasoning is used to detect complex activities.

In [1], the authors attempt to recognize robberies from streaming videos. They detect a robbery event on the basis of observations related to other suspicious events. The events that were used in robbery event detection were (i) a person running very quickly and (ii) robbing a bag all of a sudden. The robbing event consists of (i) two persons coming close to each other, (ii) then passing one another, and (iii) some object getting transferred between them. They use an ad-hoc combination of a ratio-histogram and Gaussian Mixture models to chain together the three events. The figure 2.1 is taken from [1] shows events involve in a robbery.

Figure 2.1: Example of Events Involve in a Robbery [1]

In [1], the authors have used only three events which is not sufficient to accurately detect complex activities that we consider in this thesis, while our system composes together various different kind of activities using a Bayesian framework to infer a complex activity.

Bayesian networks has previously been used for automatic inference in problems like weather prediction [55], diagnosing diseases [56], etc. We have used the Expectation Propagation algorithm [8] in our framework. In general Bayesian inference is an NP-hard problem [58]. The algorithm in [8] performs approximate Bayesian inference and has been shown experimentally [57] to be more efficient and more accurate compared to other approximate Bayesian inference algorithms like assumed-density filtering [57] for large networks.

Chapter 3

System Description for Complex Activity Recognition

Complex activities involve combinations of atomic activities. To recognize complex activities, one first needs to recognize atomic activities. Atomic activities are either articulated or non-articulated.

3.1 Video Stabilization

Full motion videos can contain translational/rotational motion of the camera that makes it difficult to track moving objects accurately in the video. So, we need to stabilize the video first before we use it for tracking and activity recognition. There are many techniques available to stabilize a video, like full frame video stabilization [24], and video stabilization using scale-invariant features [25]. The technique that we have used involves an iterative algorithm to fix the position of background pixels. This algorithm stems from the work in our group (computer vision research group) [26].

Algorithm for video stabilization

1. Find significant feature points using Shi and Tomasi's algorithm [27]. In this step, we find corner points of the objects in a frame. We compute gradients of each patch of the image in the X and Y direction. We create a matrix of the mean values of the $(X\text{-gradients})^2$, means values of the X-gradient times the Y-gradients, and mean values of $(Y\text{-gradients})^2$. If both the eigenvalues of this

matrix are non-zero then this is a feature point. We repeat this calculation for each patch in the image.

2. Use Lucas-Kanade optical flow [28] to track the moving feature points found in step 1 through each successive frame. In this step, we determine the gradients in X, Y and time (T). We try to locate feature points from one frame to next frame.
3. For each iteration, each time with a smaller bin size (half of the size of the previous iteration) for a histogram of an image translation and rotation; each bin contains a range of possible translation and rotation:
   3.1 Find the translation of each point i.e., for a set of rotations to the image we find the corresponding translation of each feature points to explain its movement from step 2.
   3.2 Compute common translational and rotational pair i.e., for each rotation and translation pair, we add one to a corresponding bin and determine which bin has the maximum value.
4. Morph the image to correct for the cumulative translational and rotational motion. We use an affine transformation (affine transformation is a combination of translation, rotation, scale, reflection, and skew) [65] with the cumulative translation and rotation values updated with values calculated in step 3.
5. The stabilized video is then fed to the object tracking module to track the location of the objects in the video.

3.2 Moving Object Tracking in Videos

Moving Objects such as cars, humans, etc. in a streaming video need to be tracked for video surveillance, robotics, etc. The object tracking module is crucial part of the complex activity recognition system. There are many object tracking algorithms available, like adaptive object tracking based on an effective appearance filter [30], object tracking by asymmetric kernel mean shift with automatic scale and orientation selection [31]. In [30], the authors provide a parametric technique for tracing moving objects in a video based on a new distance metric. This distance metric takes into account both the colors as well as the topology of a surface. In [31], the authors provide an approach for tracking moving objects in a video that adapts itself to rapidly changing motions of the camera and foreground objects. The approach in [31] is based on mean-shift; the kernel scale and orientation dynamically changes in response to changes in the foreground and the background. We have used the "agile framework for real-time visual tracking in videos" [26] to locate the position of the object in a particular frame. The agile framework is agnostic to occlusions, variations in lighting, different objects moving at different speeds, etc. It is an ensemble framework that switches between different individual algorithms based on the current state. Tracked videos are fed to the atomic activity recognition module to recognize the atomic activities in the video. The figure 3.1 is taken from [26] shows example of tracked moving objects.

Figure 3.1 : Object Tracking Example [60, 26]

3.3 Atomic Activity Detection Module

Atomic activity detection module detects atomic activities in the video, like running, walking, digging, breaking windshield, etc. Different types of objects have distinguishing motion signatures that depend on their shape and intent. Different activities are associated with distinct motion characteristics and their subtle interactions [60]. For example, non-articulated activities like vehicle turns, start and stop, human walking, running, etc. are based on the physical movement of an object across frames [60]. On the other hand, articulated activities like human digging, waving (gesturing), boxing, clapping, etc. may be associated with a stationary object where the only observed motion is that of its body parts. We have used the articulated activity analyzer to recognize the activities like breaking windshield, digging which are classified as articulated activities. The articulated activity analyzer uses a combination of techniques such as template matching (template matching involves matching part or whole of an image with another part or whole of an image) [66], horizontal and vertical histograms etc. [60] for recognizing articulated activities. Non-articulated activities like running, walking are recognized using track-based analyzer. The track-based analyzer uses the tracks generated by the tracker (tracker tracks the movement of moving objects across frames) along with a combination of algorithms including language theoretic and machine learning-based classifiers to identify activities The figures 3.2 and 3.3 show example of a non-articulated and an articulated activity respectively.

Figure 3.2: A Non-Articulated Activity

3.4 Complex Activity Recognition

The complex activity recognition module uses the atomic activities recognized by atomic activity detection module to recognize high-level plans like burglary, etc. We have used a Bayesian framework to chain the atomic activities to infer a complex activity. We discuss complex activity recognition in later part of this thesis.

Figure 3.3: An Articulated Activity [60]

Chapter 4

Reasoning about Complex Activities

4.1 Uncertainty in Knowledge

In the real world scenario, when we aim at automatically detecting complex activities like burglary, we need to handle the uncertainty in the knowledge of the environment. The uncertainty in the knowledge of the environment can be attributed to two sources: (I) the limited accuracy of the sensing mechanism [3, 4, 5, 6, 9, 10, 11] and (II) facts or activities that spatio-temporally affect a particular activity and the degree (or the probability) to which they affect it.

Causes of Uncertainty

We elaborate on the causes of the uncertainty below [3, 4, 5, 6, 9, 10, 11]:

1. Lack of Data, i.e., we don't know the complete set of causes or we may know the complete set of causes that can influence a particular activity but we cannot consider them because we don't have enough data to know their probability of occurrence. This is also called as lack of theoretical knowledge.
2. Insufficient Data, i.e., We may know the prima-facie cause for an activity and we may have the data to confirm it's truth. But we may not have enough data to estimate the probability of occurrence of that cause. This is also called as lack of practical knowledge.
3. We may be certain that we have complete set of causes and may also believe with high likelihood about the influence of each cause in the set towards a particular

activity. However freshly obtained data can produce inferences contrary to our belief. In that case a revision of the belief is needed.

In our model to detect complex activities with Bayesian inference, in order to minimize the uncertainty in the knowledge-base we have trained the framework with a large number of example videos. In our model we are required to know the possible atomic activities that can take place in a complex activity, and the sequence of those activities as well as their probability of occurrence. These are together used to compute the probability of a complex activity. In the example burglary application (refer figure 12.1), sixteen activities (man walking, ,big vehicle arrived, small vehicle arrived, exiting vehicle, breaking door, breaking window, entering building, exiting building, humans carrying some article, humans loading article into a vehicle, humans running away, humans entering into vehicle, vehicle leaving, and vehicles speed is high) are required to compute the probability of the complex activity. In figure 4.1, we have two nodes representing ways to enter a building, though there can be many ways to enter a building like entering through a pipe, but we have considered only two cases, because of lack of data.

Figure 4.1: Ways to Enter a Building

For instance, in our model we have assumed that the burglar will enter through window or door but we have not included any activity like entering through pipe which is also a possibility, as we don't have any data to confirm its occurrence.

4.2 Reasoning with Uncertainty in Knowledge

In a complex activity like burglary, there is a lot of uncertainty in the knowledge-base. Here we perform Bayesian reasoning with the uncertain knowledge, since there are two kinds of uncertainty, namely, theoretical and practical uncertainty in the activities involved in a complex activity [3, 4, 5, 6, 9, 10, 11]. Theoretical knowledge in reasoning can be defined as the knowledge about a set of causes responsible for a particular activity, i.e., the set of facts or activity that can directly influence (or deductively imply) the occurrence of a particular activity temporally or spatially e.g., uncertainty about different kinds of ways to enter in a building. Practical knowledge in reasoning is occurrence of the activities that directly influence a particular activity e.g., uncertainty about exact belief of a way to enter in a building. In figure 4.2, cause 1 and cause 2 are the possible causes for effect 1, and cause 2 and cause 3 are the possible causes for effect 2.

In our model we have represented the activities through belief networks, and have used probabilistic reasoning under uncertainty to infer the probabilities of the consequent activities given the observations. If the calculated probabilities for a complex activity are more than the preset threshold (threshold based on experimental results), then the model infers that a complex activity has occurred. Since the activities that we are focused on are suspicious and critical ones (like burglary), instead of using a MAP (Maximum A posteriori Probability) approach, we will report a suspicious activity whenever the probability of its occurrence exceeds a threshold, since the cost of missing the activity can be very high compared to the cost of a false alarm.

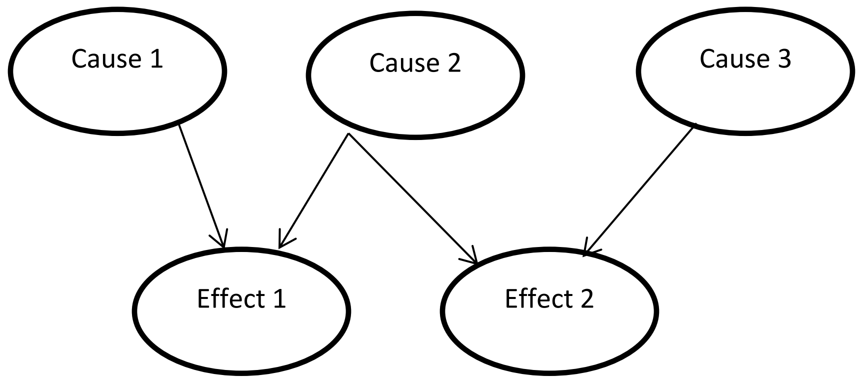


Figure 4.2: Cause Effect Relationship

4.3 Bayes Theorem [53]

The fundamental theorem on which our reasoning framework is based is the Bayes theorem.

1. Suppose E_1 is the prior event.
2. Suppose E_2 is the posterior event that depends on E_1.

   Then $P(E_2/ E_1) = P(E_1/ E_2). P(E_2)/ P(E_1)$

Figure 4.3 shows that occurrence of the event E_1 affects the probability of occurrence of the event E_2. In figure 4.3, suppose we need to find the probability of E_2 given E_1.

Then according to Bayes theorem, $P(E_2/ E_1) = P(E_1/ E_2).P(E_2)/ P(E_1)$

Where $P(E_2/ E_1)$ represents the conditional probability of E_2 given E_1; $P(E_2)$ represents probability of E_2 ; $P(E_1)$ represents probability of E_1.

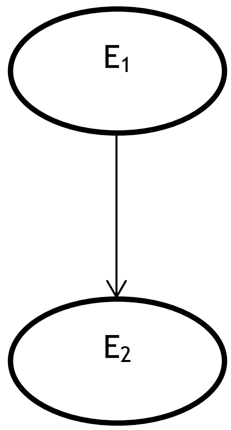


Figure 4.3: Relationship Between E_2 and E_1

4.4 Graphical Models (or Bayesian Network or Belief Network)

Probabilistic graphical models (or Belief Networks) are used for reasoning about uncertainty in knowledge. Graphical models can be seen as directed acyclic graphs. In belief networks each node is a random variable [3, 4, 5, 6, 9, 10, 11]. In our model, these random variables are possible activities that take place in complex activities. Belief networks can be used to compute the probability of a complex activity, given the probabilities of possible causes responsible for its occurrence using top-down reasoning or forward chaining. Each high level plan can be broken down into a set of activities. Each activity in turn can be decomposed into simpler activities which can be atomic or non-atomic; In top-down reasoning we explore the tree describing the hierarchy of a tree starting from leaves. Conversely it can also be used to find the probabilities of causes responsible for a complex activity when a complex activity has already been observed i.e., bottom-up reasoning [3, 4, 5, 6, 9, 10, 11, 15]. In bottom-up reasoning we explore a tree from root; we find the probabilities of complex activities given the activities and probabilities of occurrence of the atomic activities. So, we are performing top-down reasoning for complex activity recognition. Experimentally, we have seen that complex activities like burglary cannot be accurately detected using observation of just one or two activities. We consider all the atomic activities that can possibly lead to a particular complex activity, to find out the probability of possible occurrence of that activity. In a Bayesian network, for complex activity recognition, the probability for each node is computed on the basis of observed predecessor activities. For example, in the Bayesian network for complex activity detection model, in Figure 4.4, the probability of the "exiting vehicle" node depends on its predecessors, i.e., the big vehicle node and small vehicle node. Similarly probabilities of all the other nodes depend on their predecessor nodes. The prior probability and observed values are then used to calculate the posterior distributions and calculate the probability of the final composite activity.

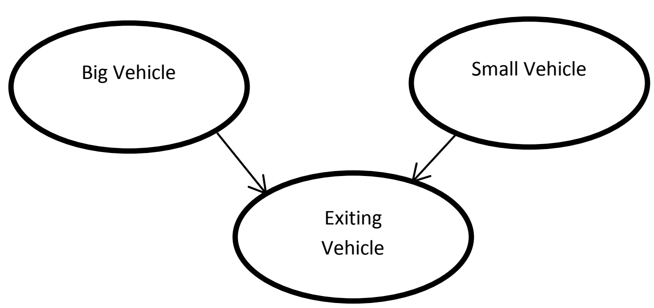


Figure 4.4: Node Dependencies

4.5 Reasoning Framework for Complex Activity Recognition

Complex Activities can involve many atomic activities. We need a framework to chain all the atomic activities together to recognize a complex activity. In figure 4.7, we present a framework for automatic recognition of complex activities of interest in an incoming full motion video stream. Surveillance videos can contain various articulated and non-articulated atomic activities. Uncertainty in surveillance videos arise due to damaged frames, lighting effect, first-time activities etc. We need to deal with both kinds of uncertainty in knowledge; theoretical as well practical knowledge uncertainty. We use belief networks to handle uncertainty in knowledge to recognize complex activity. To recognize interesting or suspicious complex activities one needs a model of normal/abnormal activities or patterns of life. For example, if we try to detect burglary, a set of common abnormal activities that we would like to detect would be: a burglar coming in car, even though no car is allowed to enter a location, or a burglar is breaking a door. We have used this abnormalcy modeling in conjunction with Bayesian reasoning to recognize interesting complex activities.

We have partitioned activities into two sets; one set has activities that take place under normal circumstances while the other has activities whose occurrence entails abnormalcy. Suppose, we want to recognize interesting complex activities for a place where big vehicles are not allowed. If the atomic activity recognition module observes a trailer, an abnormal activity is triggered. The triggering elevates the probability of suspicious complex activities. In our model, we have a Bayesian network for burglary and bayesian networks for abnormal activities. Once observations come, we feed these observations to abnormalcy model to get probabilities for abnormal activities, then we feed these observations and probabilities to Bayesian network for burglary. For e.g., In figure 4.5 and 4.6, a truck is stopping in front of a building is not an abnormal activity, but if it stops in a no loading zone, then it is an abnormal activity, and triggers the main network. Complex activity recognition can be tuned for complex activities like a burglary detection in a shop, by capturing information about how a person enters a shop, how they leave the shop, etc. In normal circumstances a person will come walking, will open the gate, enter the shop, and may leave after some time. These activities would be part of a normal activities chain. If they get off a vehicle, enter the shop by breaking a gate, come out carrying something, load articles in the car, and drive off at a high speed, these sequence of activities will be part of the set of abnormal activities. In the Bayesian framework for complex activity recognition, we have represented the activities by nodes which are random variables. The prior probability for activities that are part of abnormal activities set, are assigned a higher value, while the prior probability for activities that are part of normal activities set, are assigned a lower value. The prior distribution is the initial expectation that a particular activity will be observed.

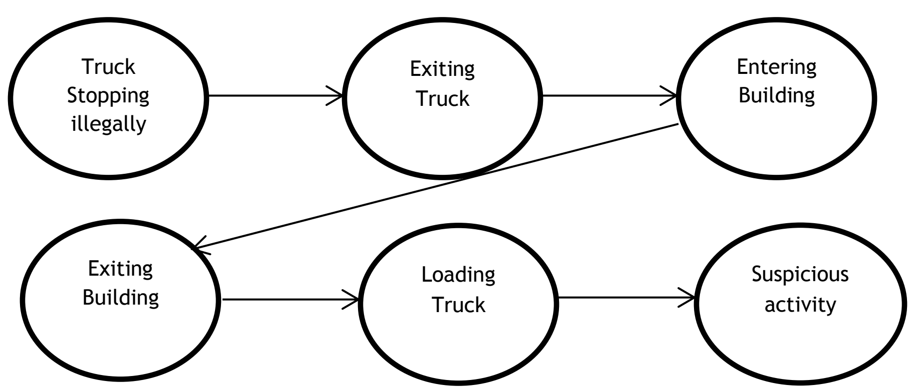


Figure 4.5: Bayesian Network

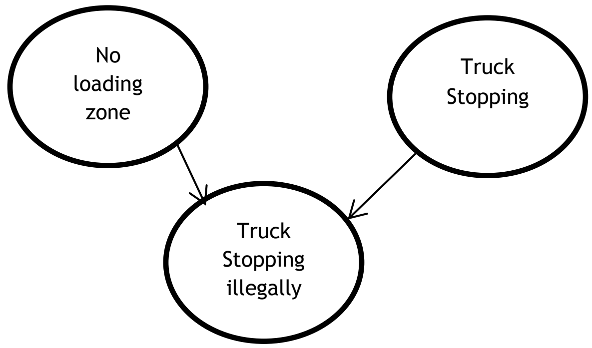


Figure 4.6: Abnormalcy Model

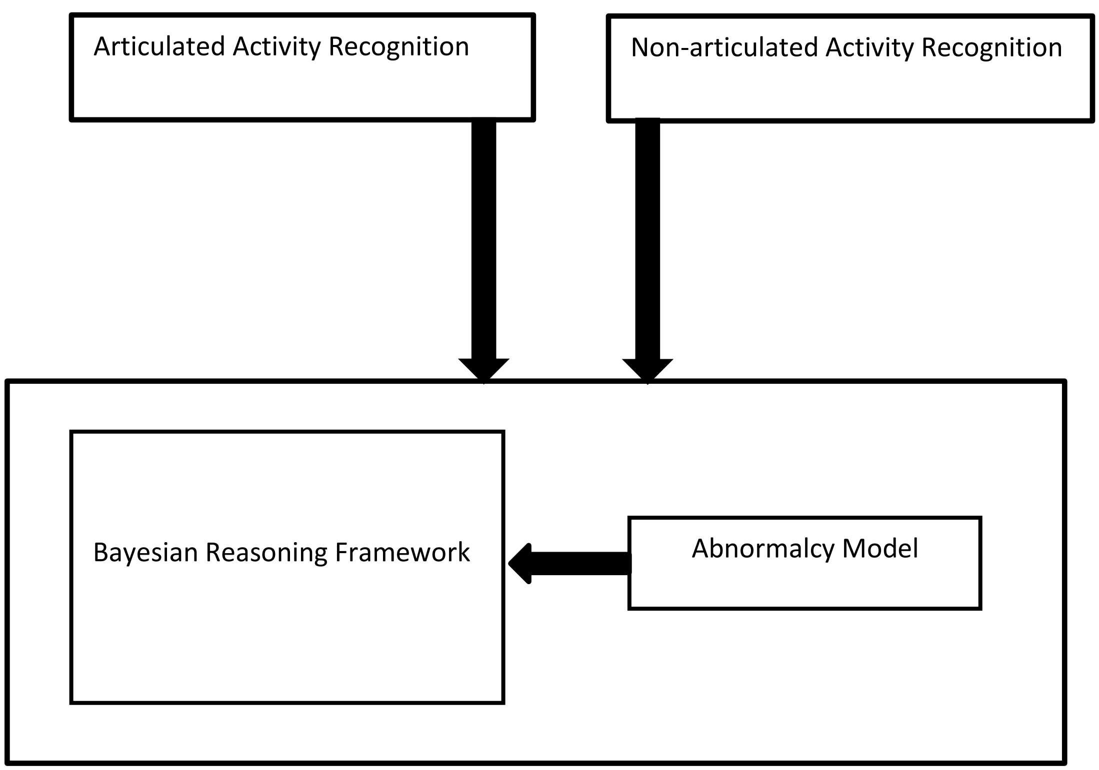


Figure 4.7: Reasoning Framework

4.6 Algorithm for Chaining Atomic Activities

Bayesian inference is a technique for inferring posterior probabilities about the occurrence of a set of events given the prior probabilities of the events (also called the belief) and a set of observations using the Bayes' theorem [9]. The events are represented as nodes in a directed acyclic graph called a Bayesian network [9]. There are many algorithms available to do inferencing in Bayesian Networks. We have used both a variable elimination algorithm [58] for exact inference and the Expectation Propagation algorithm for approximate inference in a Bayesian network to compute the probability of occurrence of a complex activity e.g., burglary. Expectation Propagation has better accuracy than other approximation algorithms available for Bayesian Inference [57]. Both variable elimination and Expectation Propagation algorithms are incorporated in our system. Depending on the size of the network, a user can invoke one of these algorithms that is more appropriate. We refer the reader to [58] for a description of the variable elimination algorithm. We describe the Expectation Propagation algorithm [57] below using figure 4.8.

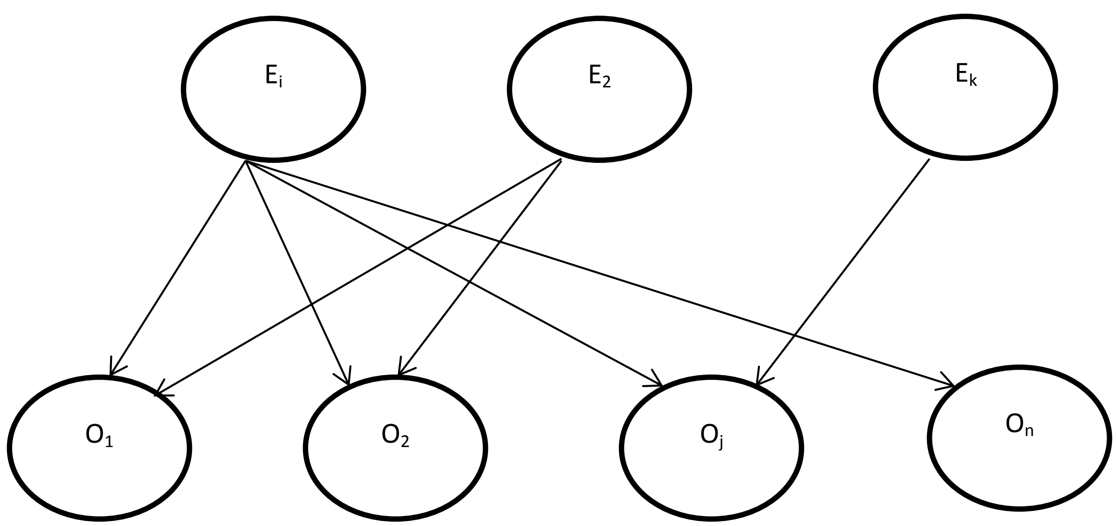


Figure 4.8: Bayesian Network to Illustrate the Expectation Propagation Algorithm

Expectation Propagation Algorithm

Start

//In figure 4.8,

$P(E_i|O_1,\ldots.,O_n) = P(E_i,O_1,\ldots.,O_n) \ / \ P(O_1,\ldots.,O_n) \propto P(E_i,O_1,\ldots.,O_n);$

//By chain rule,

$P(E_i,O_1,\ldots.,O_n) = P(O_1|\ E_{i,}\ O_2,\ldots.,O_n) \cdot P(O_2|\ E_{i,}\ O_3,\ldots.,O_n)\ldots\ldots\ldots.\ P(E_i);$

//Since in a Bayesian network a node is conditionally independent of its non-

//descendant given parents.

$P(E_i,O_1,\ldots.,O_n) = P(O_1|\ E_i) \cdot P(O_2|\ E_i)\ldots\ldots\ P(O_n|\ E_i) \cdot P(E_i);$

Let $A_1(E_i) = P(O_1|\ E_i);$

$A_2(E_i) = P(O_2|\ E_i);$

$A_n(E_i) = P(O_n|\ E_i);$

Assume an approximate distribution for $P(E_i) = 0.6 = D(E_i)$

Initialize constants $Q_1 = \ldots. = Q_n = D(E_i)$

Randomly initialize n constants $Z_1,\ldots.,Z_n \in ([0,1])$

Assume initial values of $A_1(E_i)$, $A_2(E_i),\ldots\ldots,A_n(E_i)$

Set convergence value ε

do {

$P(O_j) = P(O_j, E_i)$;

for each j=1,….,n

$P_{new}(E_i | O_j) = P(O_j|E_i) \cdot P(E_i) / P(O_j)$;

$= A_j(E_i) \cdot D(E_i) / \sum_k P(O_j|E_k) \cdot P(E_k)$;

$= A_j(E_i) \cdot D(E_i) / \sum_k A_j(E_k) \cdot D(E_k)$;

// Minimize over j the KL divergence

$KL(P(E_i|O_j))$

Let, $\alpha = \min_j KL(P(E_i|O_j), Q_j)$

And 1<=m<=n be the value of j for which the term is minimized.

retval = $P(E_i|O_m)$;

For l= 1…n;

{

$A_l(E_i) = Z_l P(E_i|O_l) / Q_l$;

$Q_l = P(E_i|O_l)$;

}

} while ($\alpha > \varepsilon$)

Return retval;

Probability Tables for figure 4.9:

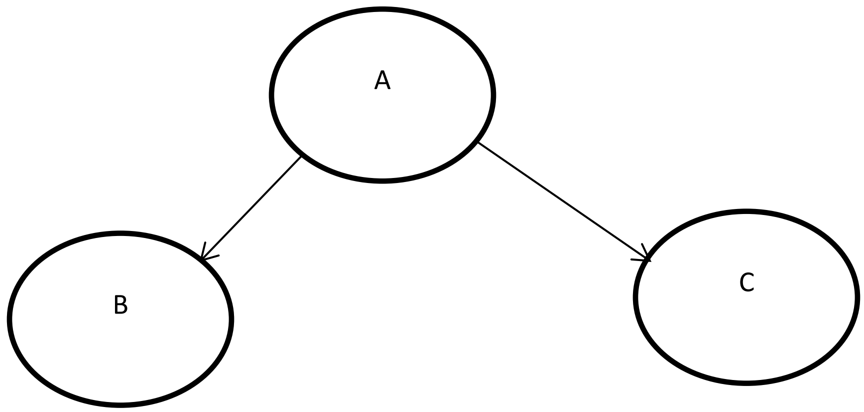


Figure 4.9: Bayesian Network Example

Table 4.1:

A=T	0.6
A=F	0.4

Table 4.2:

	A=T	A=F
B=T	0.8	0.2
B=F	0.2	0.8

Table 4.3:

	A=T	A=F
C=T	0.7	0.3
C=F	0.3	0.7

The joint probability for this network is as follows:

$P(A|B,C) = P(A) \cdot P(B|A) \cdot P(C|A)$

$P(A|B,C) = P(A) \cdot A_B(A) \cdot A_c(A)$

Assume $\varepsilon = 0.30$

Inference:

$P(A|B) = [P(B|A) \cdot P(A)] / [P(B|A) \cdot P(A) + P(B|\sim A) \cdot P(\sim A)]$

$= 0.93$

$P(A|C) = [P(C|A) \cdot P(A)] / [P(C|A) \cdot P(A) + P(C|\sim A) \cdot P(\sim A)]$

$= 0.78$

$D_{KL}(A|C) = \log(0.6/0.78) \cdot 0.6 + \log(0.4/0.22) \cdot 0.4 = 0.25$

In figure 4.7, we applied expectation propagation algorithm to this network, with the assumption that node B is true, and node C is true to infer the probability of node A; Approximate probability of node A = 0.78.

4.7 Implementation of Algorithm

1. We initialize prior probabilities for each activity based on intuition, estimates, knowledge, and experimentation. These probabilities form original ‘prior’ distribution for the model.

2. Using these priors and observed value, the compute a new probability distribution called ‘posterior’ distribution. This distribution can then be used as the ‘prior’ for another run through the model. Observed values and prior probabilities are used to calculate the posterior probabilities.
3. Finally, using more accurate posterior distributions, a final probability is calculated.

   We have used Infer.net [7] to implement a stopping complex activity recognition engine on intel I3 processor, RAM 4 GB, visual studio 2010, windows 7.

Chapter 5

Application

We have applied our complex activity recognition engine to the burglary detection problem.

5.1 Burglary Detection

Burglary is a common problem that no city or place is free from. Burglary involves many atomic activities. We need to chain all those atomic activities to infer the final burglary activity. All research so far on this topic or topics related to this has concentrated on detection of one or two activities approximately to infer a conclusion whether a burglary/robbery has taken place or not [1]. In our model we have focused on taking into consideration all the possible activities that can take place in a burglary to infer the final conclusion if a burglary has taken place or not. See figure 12.1 for the Bayesian framework that we have used for burglary detection.

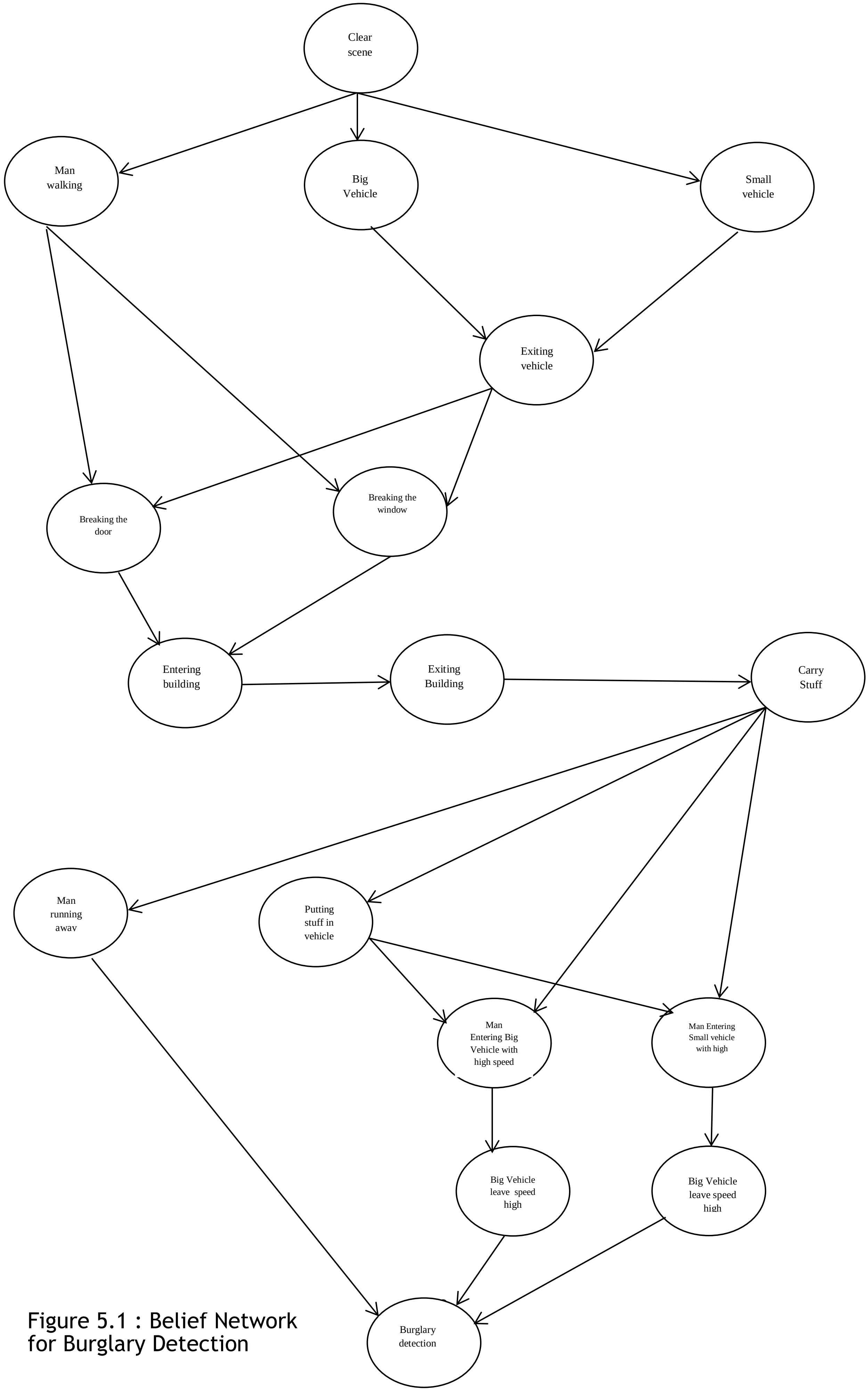


Figure 5.1 : Belief Network for Burglary Detection

Tables 5.1-5.17: Probability Tables for Burglary

Table 5.1:

Clear scene= T	0.65
Clear scene= F	0.35

Table 5.2:

	Clear Scene= T	Clear Scene= F
Man walking = T	0.6	0.4
Man walking = F	0.2	0.8

Table 5.3:

	Clear Scene= T	Clear Scene= F
Big vehicle= T	0.75	0.25
Big vehicle= F	0.3	0.7

Table 5.4:

	Clear Scene= T	Clear Scene= F
Small vehicle= T	0.7	0.33
Small vehicle= F	0.3	0.67

Table 5.5:

	Big vehicle= T, Small vehicle= T	Big vehicle= F, Small vehicle= F	Small vehicle= T, Big vehicle= F	Small vehicle= F, Big vehicle= T
Exiting vehicle = T	0.75	0.1	0.7	0.7
Exiting vehicle = F	0.25	0.9	0.3	0.3

Table 5.6:

	Man walking = T, Exiting vehicle = T	Exiting vehicle = F, Man walking = T	Exiting vehicle = T, Man walking = F	Exiting vehicle = F, Man walking = F
Open door by force= T	0.8	0.6	0.75	0.1
Open door by force= F	0.2	0.4	0.25	0.9

Table 5.7:

	Man walking = T, Exiting vehicle = T	Exiting vehicle = F, Man walking = T	Exiting vehicle = T, Man walking = F	Exiting vehicle = F, Man walking = F
Open window by force= T	0.8	0.6	0.8	0.15
Open window by force= F	0.2	0.4	0.2	0.85

Table 5.8:

	Open window by force= T, Open door by force= T	Open window by force= F, Open door by force= T	Open window by force= T, Open door by force= F	Open window by force= F, Open door by force= F
Enter building = T	0.75	0.7	0.7	0.01
Enter building = F	0.25	0.3	0.3	0.99

Table 5.9:

	Enter building = T	Enter building = F
Exit building = T	0.8	0.2
Exit building = F	0.2	0.8

Table 5.10:

	Exit building = T	Exit building = F
Carry stuff = T	0.8	0.3
Carry stuff = F	0.2	0.7

Table 5.11:

	Carry stuff = T	Carry stuff = F
Putting stuff in car = T	0.85	0.2
Putting stuff in car = F	0.15	0.8

Table 5.12:

	Putting stuff in car = T, Carry stuff = T	Putting stuff in car = F, Carry stuff = T	Putting stuff in car = T, Carry stuff = F	Putting stuff in car = F, Carry stuff = F
Running and entering big vehicle = T	0.9	0.7	0.8	0.4
Running and entering big vehicle = F	0.1	0.3	0.2	0.6

Table 5.13:

	Putting stuff in car = T, Carry stuff = T	Putting stuff in car = F, Carry stuff = T	Putting stuff in car = T, Carry stuff = F	Putting stuff in car = F, Carry stuff = F
Running and entering small vehicle = T	0.8	0.7	0.7	0.25
Running and entering small vehicle = F	0.2	0.3	0.3	0.75

Table 5.14:

	Carry stuff = T	Carry stuff = F
Running away = T	0.9	0.35
Running away = F	0.1	0.65

Table 5.15:

	Running and entering big vehicle = T	Running and entering big vehicle = F
Big vehicle leaving at high speed= T	0.8	0.4
Big vehicle leaving at high speed= F	0.2	0.6

Table 16:

	Running and entering Small vehicle = T	Running and entering Small vehicle = F
Small vehicle leaving at high speed= T	0.85	0.4
Small vehicle leaving at high speed= F	0.15	0.6

Table 5.17:

	Running away = T, Big vehicle leaving at high speed= T, Small vehicle leaving at high speed= F	Running away = F, Big vehicle leaving at high speed= T, Small vehicle leaving at high speed=F	Running away = T, Big vehicle leaving at high speed: F, Small vehicle leaving at high speed= F	Running away = F, Big vehicle leaving at high speed: F ,Small vehicle leaving at high speed= F	Running away = T, Big vehicle leaving at high speed: T ,Small vehicle leaving at high speed= T	Running away = F, Big vehicle leaving at high speed: T ,Small vehicle leaving at high speed= T	Running away = T, Big vehicle leaving at high speed: F ,Small vehicle leaving at high speed= T	Running away = F, Big vehicle leaving at high speed: F ,Small vehicle leaving at high speed= T
Burglary=T	0.7	0.75	0.7	0.1	0.8	0.7	0.7	0.7
Burglary=F	0.3	0.25	0.3	0.9	0.2	0.3	0.3	0.3

5.2 Snatching Detection

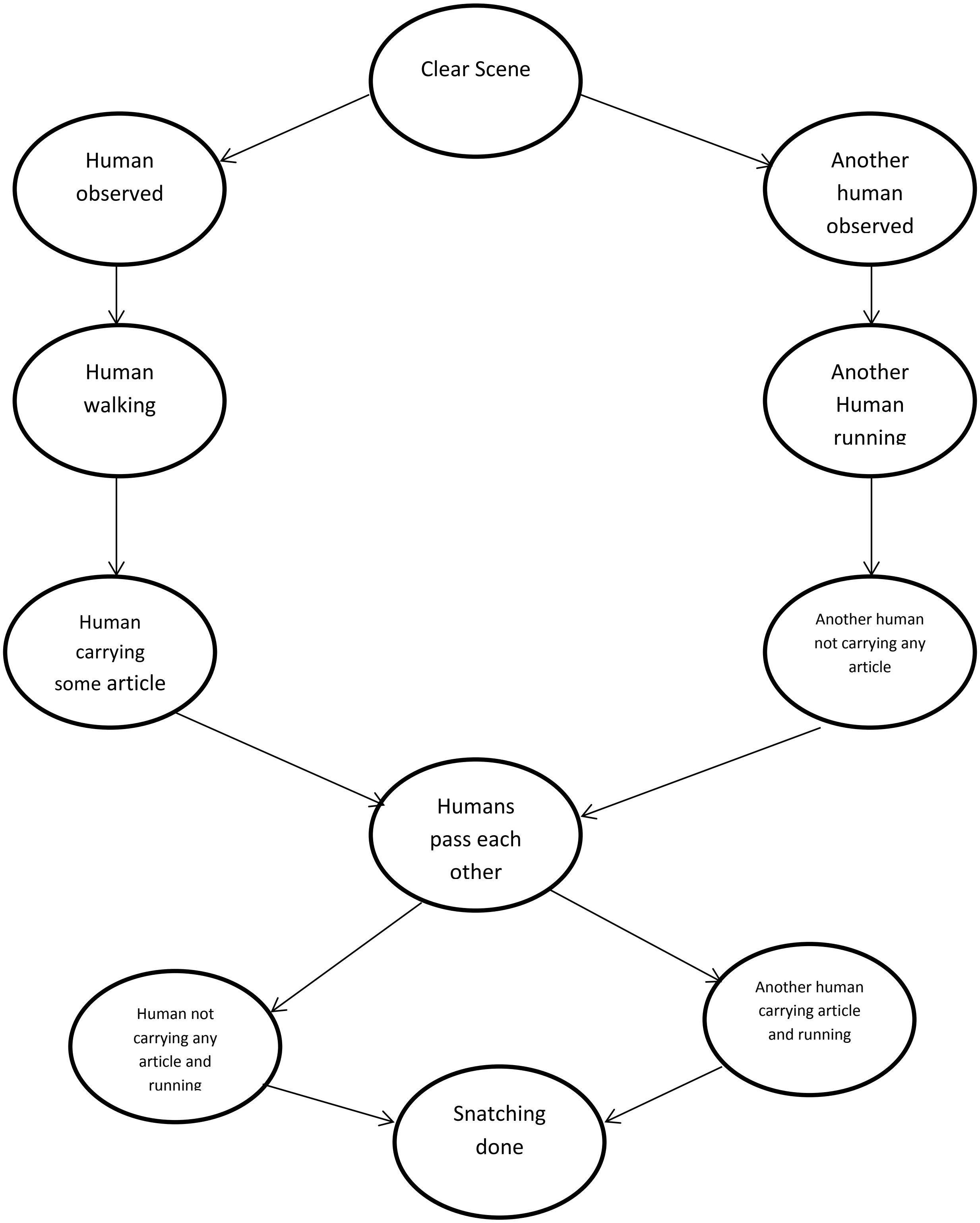


Figure 5.2 : Belief Network for Snatching

Tables 5.18-5.28: Probability Tables for Snatching

Table 5.18:

Clear scene = T	0.7
Clear scene = F	0.3

Table 5.19:

	Clear scene = T	Clear scene = F
Human observed = T	0.7	0.4
Human observed = F	0.3	0.6

Table 5.20:

	Clear scene = T	Clear scene = F
Another Human observed = T	0.7	0.35
Another Human observed = F	0.3	0.65

Table 5.21:

	Human observed = T	Human observed = F
Human walking = T	0.8	0.45
Human walking = F	0.2	0.55

Table 5.22:

	Another Human observed = T	Another Human observed = F
Human Running = T	0.7	0.35
Human Running = F	0.3	0.65

Table 5.23:

	Human walking = T	Human walking = F
Human carrying some stuff = T	0.65	0.4
Human Carrying some stuff = F	0.35	0.6

Table 5.24:

	Another Human running = T	Another Human running= F
Another Human not carrying some stuff = T	0.7	0.35
Another Human not Carrying some stuff = F	0.3	0.65

Table 5.25:

	Another Human not carrying some stuff = T, Human carrying some stuff = T	Another Human not carrying some stuff = F, Human carrying some stuff = T	Another Human not carrying some stuff = T, Human carrying some stuff = F	Another Human not carrying some stuff = F, Human carrying some stuff = F
Humans get into contact = T	0.7	0.25	0.65	0.2
Humans get into contact = F	0.3	0.75	0.35	0.8

Table 5.26:

	Humans get into contact = T	Humans get into contact = F
Human not carrying some stuff and running = T	0.9	0.3
Human not carrying some stuff and running = F	0.1	0.7

Table 5.27:

	Humans get into contact = T	Humans get into contact = F
Another Human carrying some stuff and running = T	0.85	0.2
Another Human carrying some stuff and running = F	0.15	0.8

Table 5.28:

	Another Human carrying some stuff = T, Human not carrying some stuff = T	Another Human carrying some stuff = F, Human not carrying some stuff = T	Another Human carrying some stuff = T, Human not carrying some stuff = F	Another Human carrying some stuff = F, Human not carrying some stuff = F
Snatching = T	0.99	0.6	0.65	0.4
Snatching = F	0.01	0.4	0.35	0.6

5.3 Dataset

We have taken video 1, video 2, video 3, video 4, video 5, video 6, video 7, video 8, video 15, video 16, video 17, video 18, and video 19 from youtube [62], and video 9, video 10, video 11, video 12, video 13, video 14, video 20, and video 21 from virat [64] data set.

5.4 Results

Here we have taken the snapshots from the videos on which the burglary detection module is applied (See appendix: snapshots), this will give the better understanding of what activities in general takes place in a burglary. Full length tracked results are available at [63], and snapshots of videos are given at the end.

5.5 Analysis

We have analyzed the results using a confusion matrix and a ROC curve. Refer figure 5.3, 5.4, 5.5, 5.6, and 5.7.

5.5.1 Confusion Matrix

Confusion Matrix is used to show the result or performance of classification system in terms of actual case versus predicted case. We have also used this confusion matrix to analyze the performance in terms of the classification model's ability to classify the negative and positive cases. Here true positive indicates that complex activity is taken place, and it is correctly classified as a positive case ; false positive indicates that a complex activity is not taken place but model wrongly classify this as a positive case; false negative indicates that a complex activity is taken place but model wrongly classify this activity as a negative case; and true negative indicates a complex activity is not taken place and model correctly classify this complex activity as a negative case. [12]

5.5.2 Receiver Operating Characteristic (ROC) Curve

This Receiver Operating Characteristic curve is a graph which is used to analyze the result or performance of the classification model on application of dataset on it. In the ROC curve the X axis represents false positive rate and Y axis represents true positive rate. The ROC curve is based on all possible thresholds.[13]

Predicted / Actual	Burglary Took Place	Burglary Not Took Place
Burglary Took Place	**6**	**0**
Burglary Not Took Place	**4**	**4**

Figure 5.3: Confusion Matrix for Burglary

True Positives 6 (Actual Burglary That was classified as Burglary)	**False Negatives 0 (Actual Burglary That was wrongly classified as No Burglary)**
False Positives 4 (No Actual Burglary That was wrongly classified as Burglary)	**True negatives 4 (No Actual Burglary That was classified as No Burglary)**

Figure 5.4: Confusion Matrix for Burglary

Actual \ Predicted	Snatching Took Place	Snatching Not Took Place
Snatching Took Place	**4**	**0**
Snatching Not Took Place	**0**	**3**

Figure 5.5: Confusion Matrix for snatching

True Positives 4 **(Actual Snatching that was classified as Snatching)**	**False Negatives 0** **(Actual Snatching that was wrongly classified as No Burglary)**
False Positives 0 **(No Actual Snatching that was wrongly classified as Snatching)**	**True negatives 3** **(No Actual Snatching that was classified as No Snatching)**

Figure 5.6: confusion Matrix for Snatching

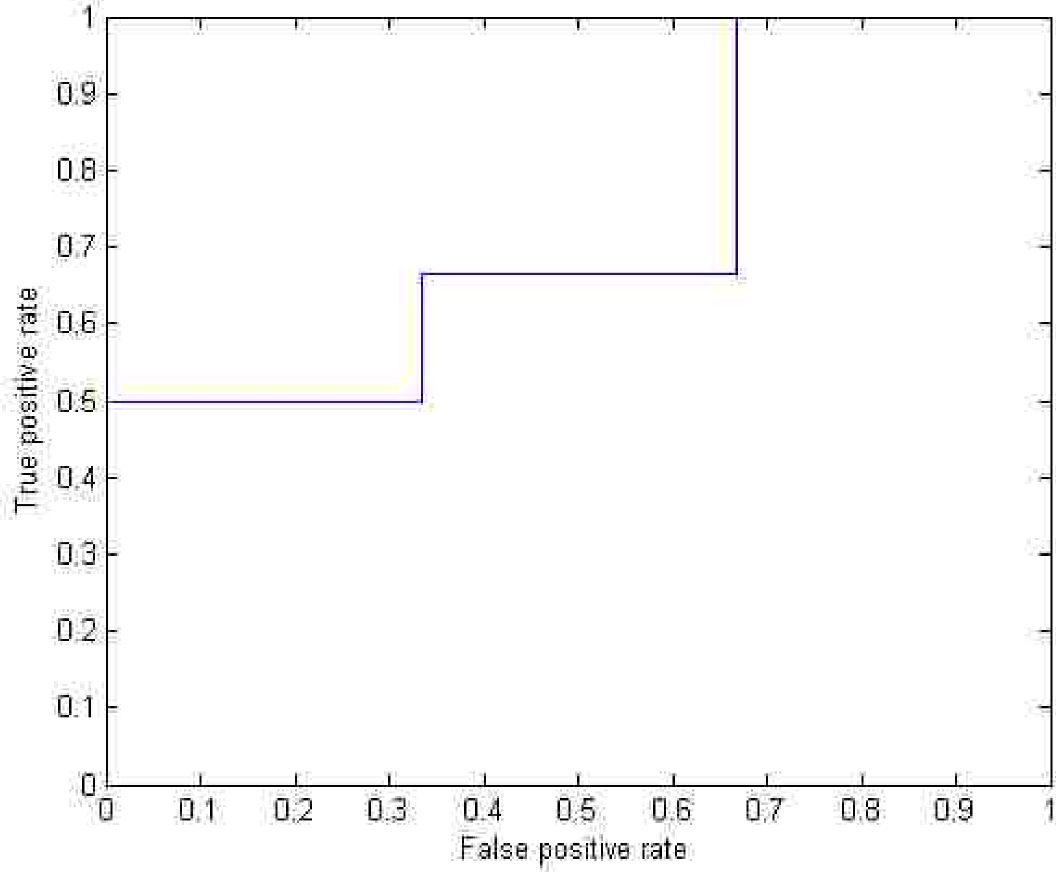


Figure 5.7: ROC Curve

5.6 Timing Data

Burglary:

Source	Original video duration	Processing time by approximate inference	Processing time by exact inference
Youtube: Video 1	230 seconds	2.54 seconds	1.01 seconds
Youtube: Video 2	73 seconds	2.34 seconds	0.93 seconds
Youtube: Video 3	180 seconds	2.51 seconds	0.99 seconds
Youtube: Video 4	101 seconds	2.61 seconds	0.81 seconds
Youtube: Video 5	55 seconds	1.93 seconds	1.03 seconds
Youtube: Video 6	403 seconds	2.21 seconds	0.89 seconds
Youtube: Video 7	244 seconds	2.71 seconds	0.86 seconds
Youtube: Video 8	133 seconds	2.61 seconds	1.07 seconds
VIRAT: Video 9	43 seconds	2.01 seconds	0.86 seconds
VIRAT: Video 10	31 seconds	2.16 seconds	1.11 seconds
VIRAT: Video 11	63 seconds	2.23 seconds	0.96 seconds
VIRAT: Video 12	173 seconds	2.43 seconds	0.86 seconds
VIRAT: Video 13	58 seconds	2.47 seconds	1.06 seconds
VIRAT: Video 14	83 seconds	2.56 seconds	0.83 seconds

Snatching:

Source	Original video duration	Processing time by approximate inference	Processing time by exact inference
Youtube: Video 15	12 seconds	2.11 seconds	0.86 seconds
Youtube: Video 16	25 seconds	1.94 seconds	0.91 seconds
Youtube: Video 17	10 seconds	1.98 seconds	0.68 seconds
Youtube: Video 18	13 seconds	2.03 seconds	0.79seconds
Youtube: Video 19	18 seconds	2.12 seconds	0.84 seconds
VIRAT: Video 20	20 seconds	2.27 seconds	0.91 seconds
VIRAT: Video 21	23 seconds	2.16 seconds	0.77 seconds

Chapter 6

Conclusions and Future Work

6.1 Conclusion

We have developed a complex activity recognition engine to detect composite complex activities with applications to burglary detection. This system takes as input a streamed video, which is then passed to the video stabilizer for preprocessing [26]. The video stabilizer corrects the damaged frames. The preprocessed video is then passed to atomic activity recognition module to recognize the articulated and non-articulated activities. The articulated and non-articulated activities recognized by atomic activity recognition engine are then used by complex activity recognition to recognize complex activities like burglary which involves composition of many articulated and non-articulated activities. The complex activity recognition engine uses a Bayesian reasoning framework in conjunction with abnormalcy models to recognize the complex activity. Complex activity recognition has a wide variety of applications in the fields of surveillance, battle field, robotics, etc. Our engine can recognize complex activities in real time. Results depend a lot on the context, the location, as well as the region of interest; For example, if a video is taken from a parking lot, then to get more accurate results we should not use the same prior probabilities that we use for a bank, or an antique shop, etc. We can also get accurate results by increasing/decreasing the threshold as per the region of interest. While assigning prior probabilities we also need to take abnormal activities into calculation under the given circumstances.

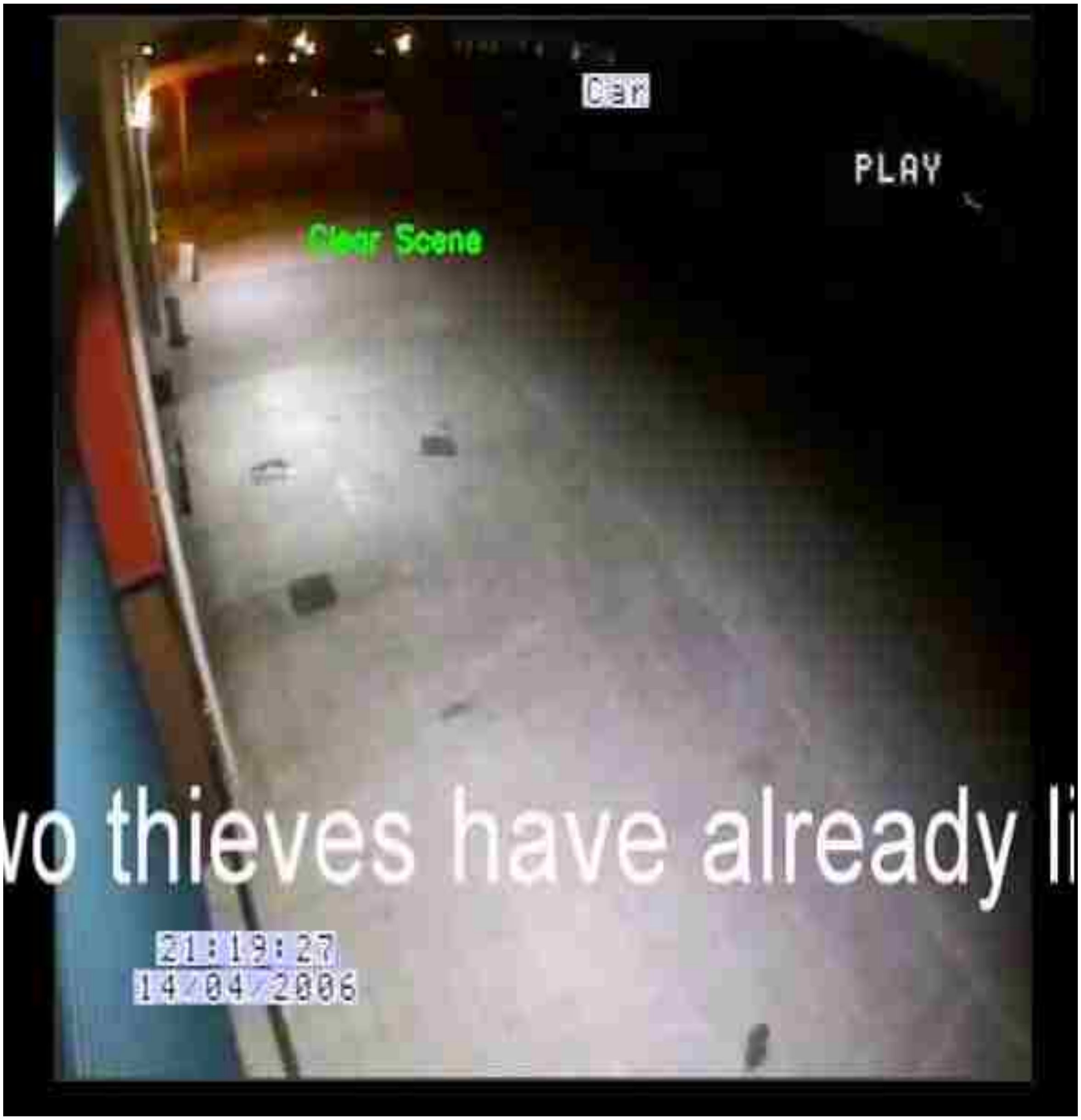
6.2 Future Work

We are working on improving, and extending this complex activity recognition system to recognize other complex activities. We are also working on using sophisticated machine learning techniques to improve the accuracy. One such technique is deep learning. Deep learning [44] involves learning level by level [46], e.g., we can get more accurate results in complex activity recognition using deep learning technique as we can make use of many more details, for example, for atomic activity like big vehicle is arrived, there can be more details like size, type which can then be used for further classification, and the prior probabilities can be set using knowledge obtained from prior observed complex activities. In deep learning, system first learns the lowest level and then goes to learn next level up in the hierarchy and keeps doing this up to the highest level. In [48], the authors have developed a vision system using deep learning; They tested it on robot vision problems and found that deep learning can be used to obtain accurate results in object classification.

Appendix: Snapshots

Snapshots for video 1: The following video was taken from public dataset youtube, it was true positive case. The location of the video was at a place where big vehicles were not allowed, so here big vehicle, breaking the door, etc. were abnormal activities. In this video, many articulated and non-articulated activities were observed. Observed activities were big vehicle arrived (activity that belongs to abnormal activities set), humans came out of this big vehicle, then they were observed breaking a door (activity that belongs to abnormal activities set), carrying some stuff, putting stuff in a big vehicle, entering a vehicle at a high speed, and finally big vehicle was observed leaving the place at a high speed. We found that high probability of burglary was detected for this video.

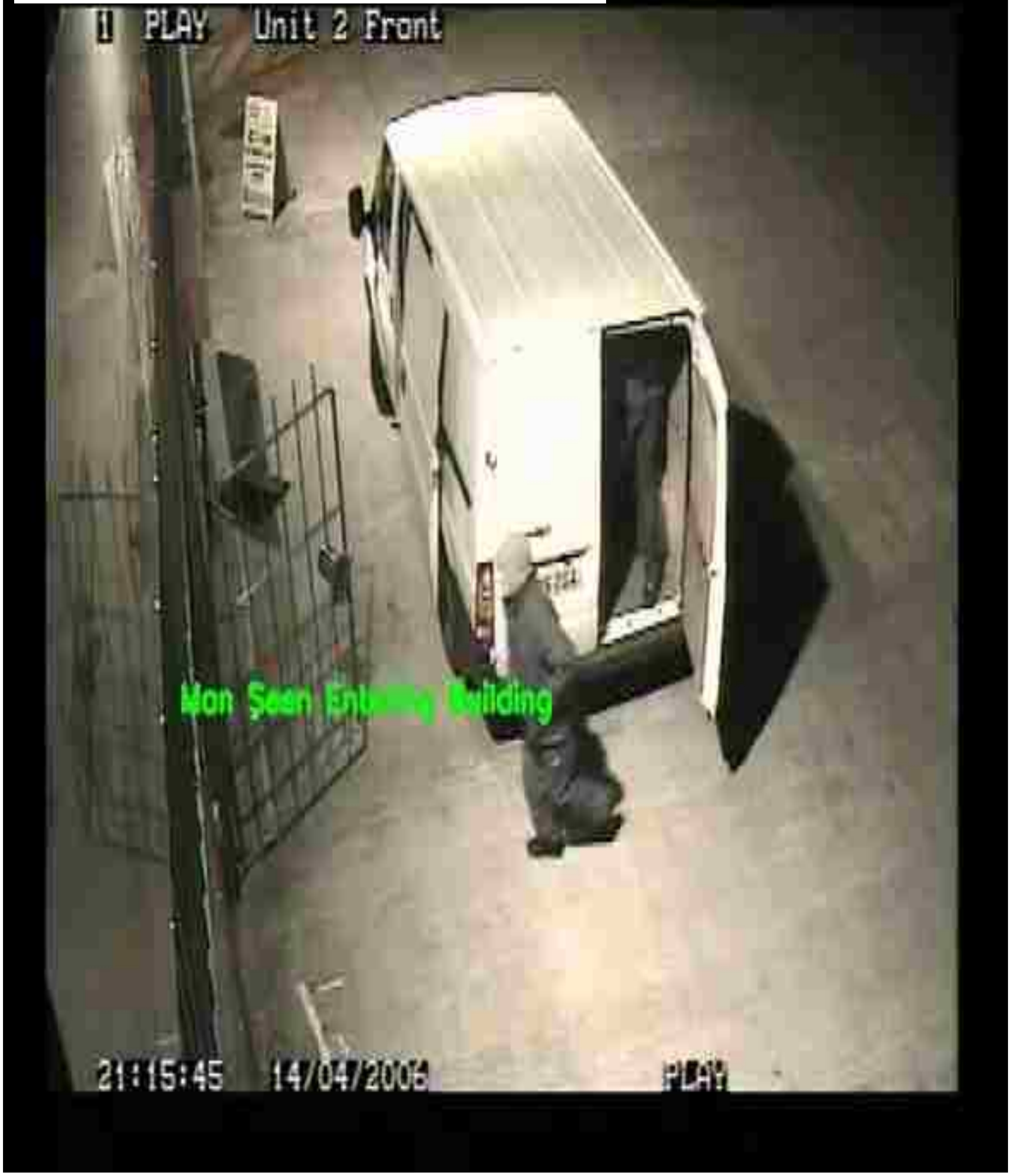
It is observed that scene is clear.



It is observed that big vehicle is

It is observed that some humans are breaking a door

It is observed that human are entering into a building

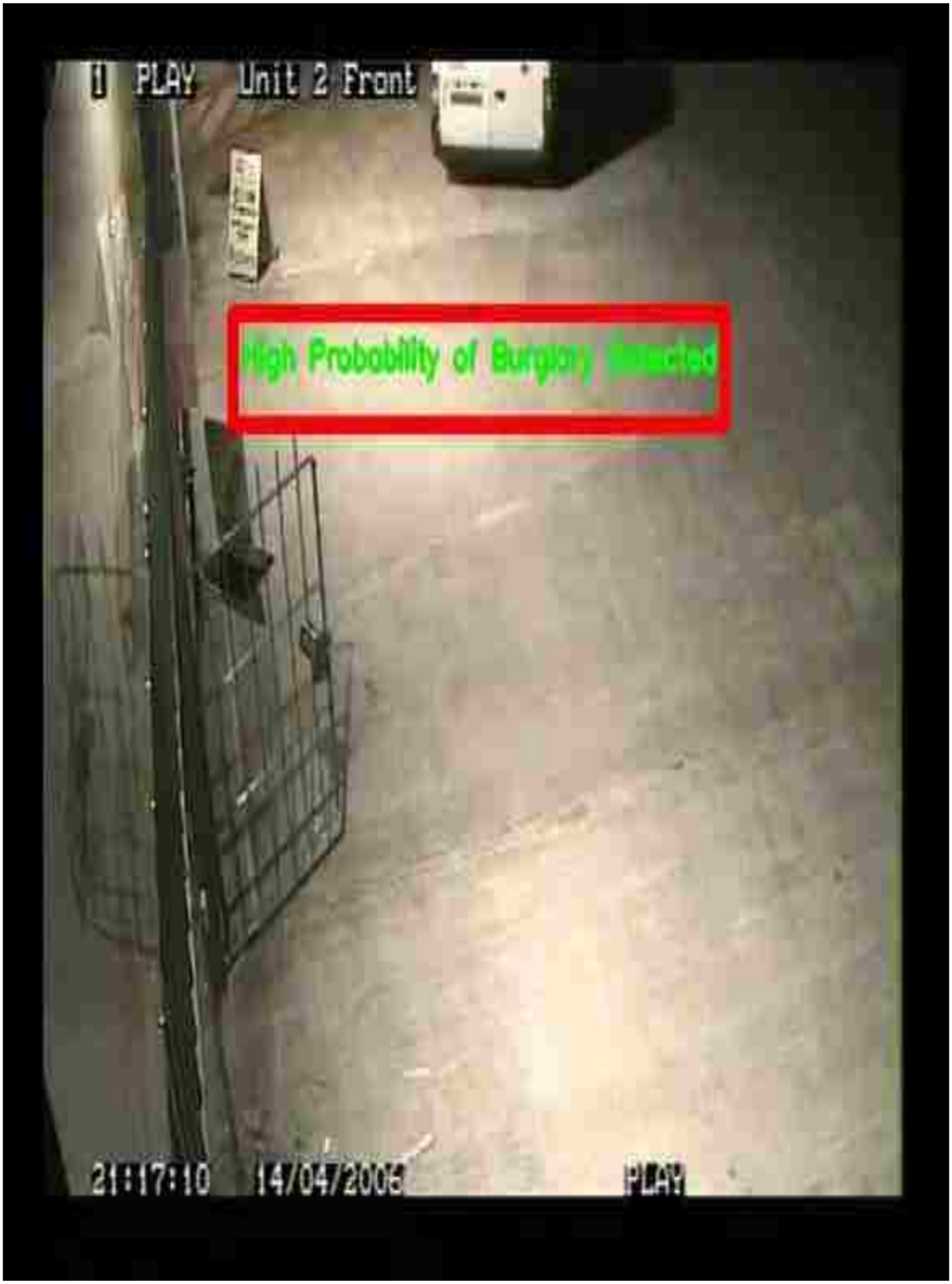


It is observed that humans are carrying some

It is observed that humans are loading articles into a big vehicle

It is observed that human are running and entering into a big vehicle

It is observed that big vehicle is leaving and speed is high


1 PLAY Unit 2 Front
Probability of Burglary
21:17:10 14/04/2006
PLAY

Snapshots for Video 2: The following video was also taken from youtube. Here breaking door, people running away, were abnormal activities. In this video, at first some humans were observed, then humans were seen breaking a door (abnormal activity), entering the building, and finally running away (abnormal activity). So, high probability of burglary was detected for this video. It was a true positive case.


It is observed that humans are walking
Man Walking

It is observed that more humans are walking.

It is observed that humans are breaking a door.

It is observed that human are entering into a building

It is observed that a human is running away.

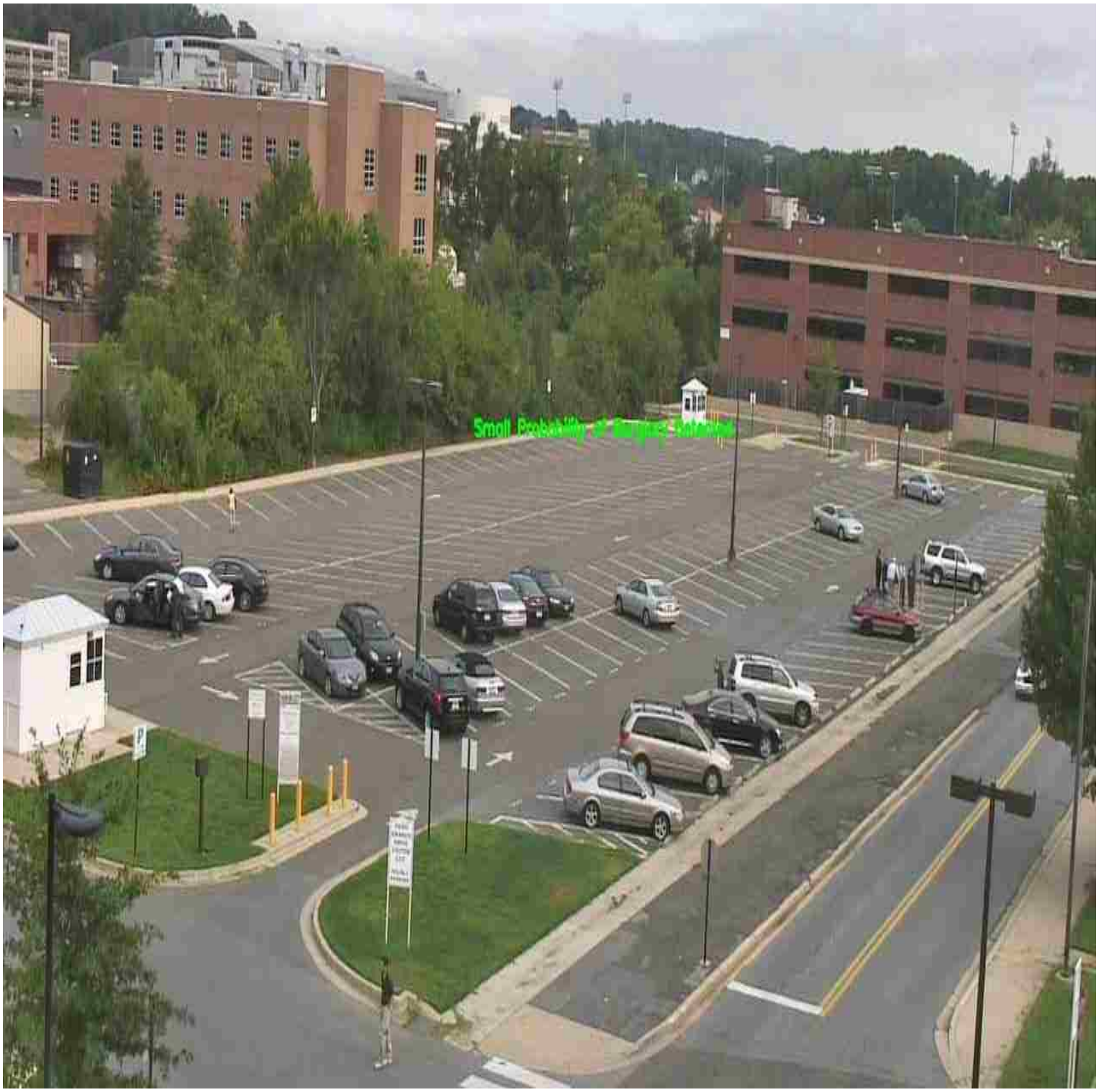

Anyone with information contact the HCSO
at 423-272-4848.

Snapshots for video 3: The following video was taken VIRAT dataset. This video is taken from parking lot. Here low probability of burglary was detected. It was a true negative case.

It is observed that a human is walking.

A big vehicle is observed.

It is observed that a human is walking.

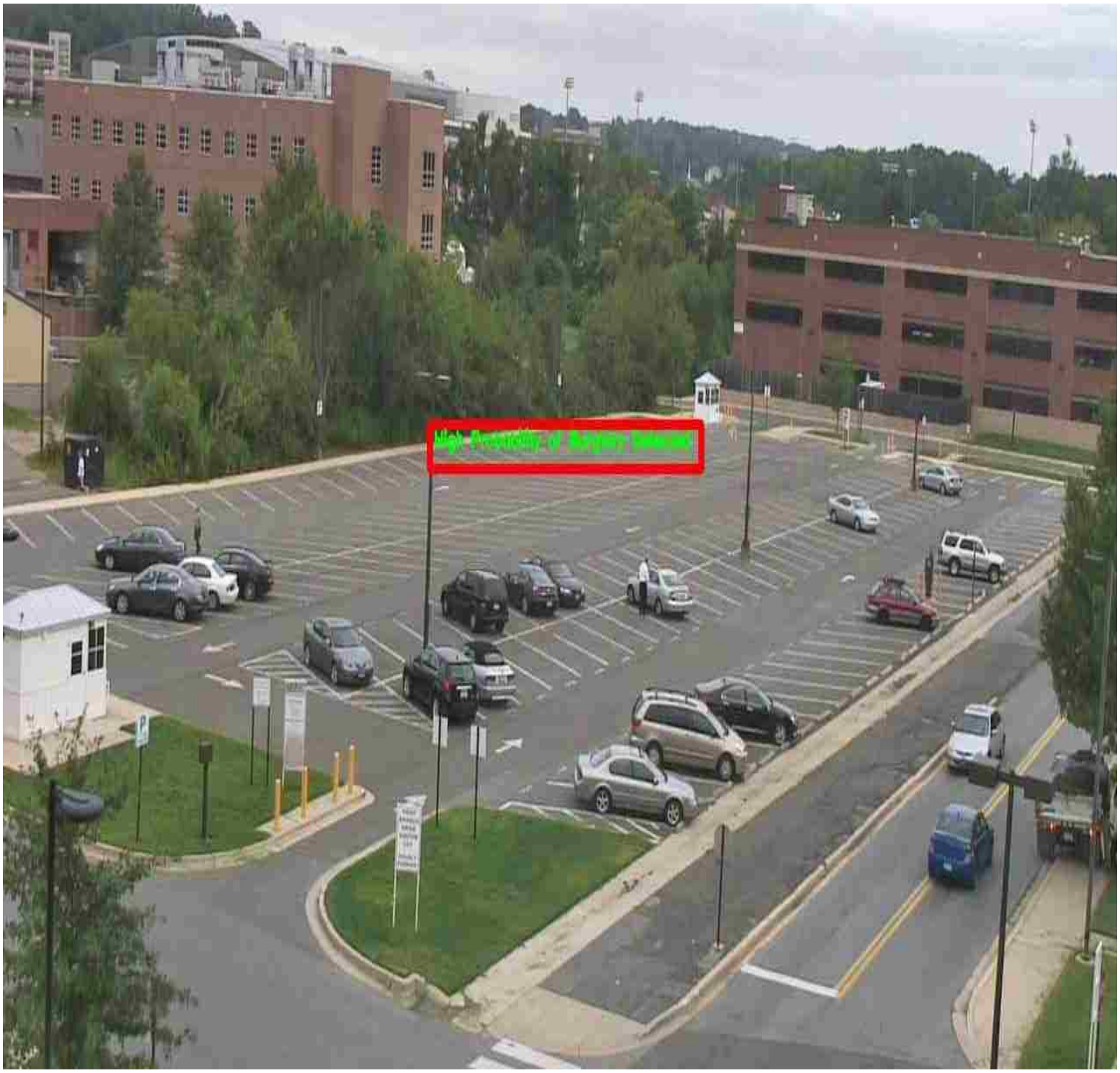

Small Probability

A big vehicle is observed.

Another big vehicle is observed.

It is observed that a human is coming out of a big vehicle.

It is observed that a big vehicle is leaving and speed is high.



Bibliography

[1] Chuang, C. H., Hsieh, J.W., Fan, K.C., 2007. Suspicious Object Detection and Robbery Event Analysis. Computer Communications and Networks, 2007.

[2] Cowell, R. (1998). An introduction to inference for Bayesian networks. In (Jordan, 1998), pages 9–26.

[3] www.cs.uic.edu/~liub/teach/cs511.../cs511-uncertainty.doc (https://docs.google.com/viewer?a=v&q=cache:pFpzW6BZT6YJ:www.cs.uic.edu/~liub/teach/cs511-spring-06/cs511-uncertainty.doc+&hl=en&gl=us&pid=bl&srcid=ADGEESjPl_-JqtmNVzv8KUJqhFDmbbICY2A-ZGUhqYte7OVDvnbfErutdmy-pIhSQJZBnqHGtniUYVpkOVBEmlmOfq0m81DITcBE1JRlc0scOhsuf9-cG9LVSecurG2BxPBCs7JTOWKz&sig=AHIEtbQRu36EqDJN5movLxtFMxJrPBCW2Q)

[4] http://www.youtube.com/watch?v=1BRIjhX4JdU&feature=relmfu

[5] http://www.youtube.com/watch?v=rFQsbArQE6Y

[6] http://www.youtube.com/watch?v=cMN6ykIYF_U

[7] T. Minka, J. Winn, J. Guiver, and D. Knowles Infer.NET 2.5, Microsoft Research Cambridge, 2012. http://research.microsoft.com/infernet

[8] T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kauffmann, 2001.

[9] **http://en.wikipedia.org/wiki/Bayesian_inference**

[10] **http://en.wikipedia.org/wiki/Bayesian_network**

[11] **http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html**

[12] **http://en.wikipedia.org/wiki/Confusion_matrix**

[13] **http://en.wikipedia.org/wiki/Receiver_operating_characteristic**

[14] Giordano, L. ,Martelli, A., Schwind, C., 2001. Reasoning about actions in dynamic linear time temporal logic, in: FAPR'00—Int. Conf. on Pure and Applied Practical Reasoning, London, September 2000, Logic J. IGPL 9 (2) 289–303.

[15] **http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html**

[16] Kreutzmann, A. , Colonius, I., Frommberger, L. , Dylla, F., Freksa, C. , Wolter, D. , 2011. On Process Recognition by Logical Inference Proceedings of the European Conference on Mobile Robots.

[17] Junejo, I., Dexter, E., Laptev, I., P´erez, P.,2008. Cross-view action recognition fromtemporal self-similarities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306.

[18] Lucas, B. D., Kanade, T., 1981. An Iterative Image Registration Technique With An Application To Stereo Vision. Proc. Seventh International Joint Conference on Artificial Intelligence (IJCAI-81), Vancouver, pages 674-679.

[19] Peirce, C. S., 1901. On the Logic of drawing History from Ancient Documents especially from Testimonies. Collected Papers v. 7, paragraph 219.

[20] Pnueli, A., 1977. The temporal logic of programs. Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS), 46–57.

[21] Poole, D., 1993. Logic Programming, Abduction and Probability: a top-down anytime algorithm for estimating prior and posterior probabilities. New Generation Computing, vol. 11, no. 3-4, pp. 377-400.

[22] Raghavan, S. Mooney, R. J. ,2011. Abductive Plan Recognition by Extending Bayesian Logic Programs, In Proc. Of ECML/PKDD.

[23] Zhou, Q. , Aggarwal, J. K., 2006. Object tracking in an outdoor environment using fusion of feature and cameras. Image and Vision Comp., 24(11):1244–1255.

[24] Y. Matsushita, E. Ofek, X. Tang, and H.Y. Shum, "Full-Frame Video Stabilization," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 50-57, 2005.

[25] R. Hu, R. Shi, I Shen, and W.Chen, "Video Stabilization using scale-invariant features" *Proceedings of the 11th International Conference Information Visualization* ,pp**.** 871-877 , 2007

[26] S. Basu, M. Stagg, R. DiBiano, M. Karki, S. Mukhopadhayay, and J. Weltman, "An agile framework for Real-Time Visual Tracking in Videos" Working Paper.

[27] J. Shi and C. Tomasi, "Good Features To Track", *9th IEEE Conference on Computer Vision and Pattern Recognition*, pages- 593-600, 1994.

[28] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique With An Application To Stereo Vision", *Proc. Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, Vancouver. pages 674-679, 1981.

[29] http://en.wikipedia.org/wiki/Video_tracking

[30] Wang, H., et al., Adaptive object tracking based on an effective appearance filter. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. 29(9): p. 1661-1667.

[31] A. Yilmaz, “Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1-6, June 2007.

[32] M. S. Ryoo and J. K. Aggarwal, ”Recognition of High-level Group Activities Based on Activities of Individual Members”, Proceedings of IEEE Workshop on Motion and Video Computing (WMVC), Copper Mountain, CO, January 2008.

[33] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. IEEE T CSVT, 18(11):1473–1488, Nov 2008.

[34] Hagar, G., Belhumeur, P.: Efficient Region Tracking With Parametric Models of Geometry and Illumination, IEEE Transactions on Pattern Analysis and Machine Intel- ligence Vol. 20 No. 10: pp. 1125-1139.

[35] P. Kaewtrakulpong and R. Bowden, “An Improved Adaptive Background Mixture Model For Real-Time Tracking With Shadow Detection”, Proc. European Workshop Advanced Video Based Surveillance Systems, 2001.

[36] C. Chang, R. Ansari, and A. Khokhar, “Multiple object tracking with kernel particle filter”, in Proc. CVPR, 2005, pp. 566-573.

[37] H. Tao, H.S. Sawhney, and R. Kumar, “Object Tracking with Bayesian Estimation of Dynamic Layer Representations,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 75-89, Jan. 2002.

[38] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In Proc. Int’l Conf. Computer Vision, pages 353–360, Nice, France, 2003.

[39] Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: A survey. Circuits and Systems for Video Technology, IEEE Transactions on 18 (2008) 1473-1488.

[40] Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Comput. Surv. 43 (2011) 16:1-16:43.

[41] Hoogs, A., Perera, A.G.A.: Video activity recognition in the real world. In: Proceedings of AAAI. (2008).

[42] Li, B., Ayazoglu, M., Mao, T., Camps, O., Sznaier, M.: Activity recognition using dynamic subspace angles. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. (2011) 3193-3200.

[43] Choi, W., Shahid, K., Savarese, S.: Learning context for collective activity recognition.In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. (2011) 3273-3280.

[44] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—A new frontier in artificial intelligence research," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, 2010.

[45] http://en.wikipedia.org/wiki/Deep_learning

[46] http://deeplearningworkshopnips2011.files.wordpress.com/2012/01/workshop_tutorial_part1.pdf

[47] Lauzon, F.Q.; , "An introduction to deep learning," *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on* , vol., no., pp.1438-1439, 2-5 July 2012 doi: 10.1109/ISSPA.2012.6310529

[48] Hadsell, R.; Erkan, A.; Sermanet, P.; Scoffier, M.; Muller, U.; Yann LeCun; , "Deep belief net learning in a long-range vision system for autonomous off-road driving," *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* , vol., no., pp.628-633, 22-26 Sept. 2008 doi: 10.1109/IROS.2008.4651217

[49] Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: A survey. Circuits and Systems for Video Technology, IEEE Transactions on 18 (2008) 1473 {1488

[50] Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Comput. Surv.43 (2011) 16:1-16:43

[51] Shah, M., Javed, O., Sha_que, K.: Automated visual surveillance in realistic scenarios. IEEE MultiMedia 14 (2007) 30-39

[52] http://www.sci.utah.edu/~gerig/CS6640-F2010/prob-tut.pdf

[53] **http://en.wikipedia.org/wiki/Bayes'_theorem**

[54] **http://www.computing.dcu.ie/~pgunning/Axioms.pdf**

[55] Nandar, A.; , "Bayesian network probability model for weather prediction," *Current Trends in Information Technology (CTIT), 2009 International Conference on the* , vol., no., pp.1-5, 15-16 Dec. 2009 doi: 10.1109/CTIT.2009.5423132

[56] Fesharaki, Nooshin Jafari; Pourghassem, Hossein; , "Medical X-ray Images Classification Based on Shape Features and Bayesian Rule," *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on* , vol., no., pp.369-373, 3-5 Nov. 2012 doi: 10.1109/CICN.2012.145

[57] Minka, T. P. (2001). *A family of algorithms for approximate Bayesian inference*. Doctoral dissertation, Massachusetts Institute of Technology.

[58] A. Dawiche; "Modeling and reasoning with Bayesian networks," Cambridge University Press; 1 edition (April 6, 2009).

[59] Dalal, N.; Triggs, B.; , "Histograms of oriented gradients for human detection," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on , vol.1, no., pp.886-893 vol. 1, 25-25 June 2005 doi: 10.1109/CVPR.2005.177

[60] S. Basu, R. DiBiano, M. Stagg, M. karki, S. Mukhopadhyay, I. Bhale "A Symbolic Framework for Recognizing Activities in Full Motion Surveillance Videos", Working Paper.

[61] S. Mukhopadhyay, I. Bhale, S. Basu, R. DiBiano, M. Stagg, M. karki, "Plan recognition in surveillance videos using abductive reasoning on linear temporal logics" under review, International Conference on Acoustics, Speech, and Signal Processing 2013.

[62] http://youtube.com

[63] https://tigerbytes2.lsu.edu/users/ibhale1/videos/

[64] http://www.viratdata.org/

[65] http://en.wikipedia.org/wiki/Affine_transformation

[66] http://en.wikipedia.org/wiki/Template_matching

Vita

Ishan Bhale was born in Gwalior, Madhya Pradesh, India. He finished his schooling from A.M.I. Shishu Mandir Higher Secondry School, Gwalior in March, 2006. He earned his Bachelor of Engineering (B.E.) degree from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal in June, 2010. After finishing his engineering, he joined Louisiana State University (LSU) to pursue Master of Science in Systems Science. He completed his Master of science in Systems Science from LSU in May, 2013.