CSE 591 Intelligent Assistive Robotics
Smart Video Surveillance System

Chirav Dave, Diptanshu Purwar, Natasha Mittal
{cdave1, dpurwar, nmittal4}@asu.edu

1 Introduction

An important problem at the center of artificial intelligence is the design of suitable control policies for intelligent systems[12].Video surveillance systems help to enhance security and safety of any place where they are installed and thus are being increasingly used in public places such as banks, airports and shopping malls as well as in confidential and critical settings like government and military premises. The increasing availability and lower cost of higher quality cameras lead to multiple camera installations for monitoring target areas. In this paper, we present a smart video surveillance system for determining the optimal course of actions that can be taken by a camera for monitoring the movement of an intruder in a real time setting. Our proposed intelligent system guides the camera by estimating the direction in which the intruder is currently moving, as well as the discretized intruder's location corresponding to different frames in the video. We used tiny YOLO[13] for capturing intruders location and direction in the video and a Partially Observable Markov Decision Process (POMDP)[1][11] formulation to solve the decision-making problem of tracking an intruder based on noisy estimates of the object location and object direction. We used a POMDP solver[10] to compute a policy that balances the cost of directing the camera and tracking the person of interest with the benefit of the camera to continuously monitor a target area for possible intrusion detection.

1.1 Motivation

Automatic camera control via video surveillance is an interesting research area. By automatic camera control, we imply, that if an intruder is suspected, then the camera is required to track a person continuously within the observed area. The system should be able to predict possible trajectories of the intruder based on estimates of intruder behaviour. Our goal in this project is to develop an intelligent video surveillance system so that it can automatically detect malicious activities in the area under surveillance and raise alarm. Inspired by the advancement in deep neural networks, we chose the tiny YOLO[13] network which is the state-of-the-art in object detection and recognition. Also, since the camera would generate a high volume of graphic data, we would have to process this data as quickly as possible to achieve real-time performance[11]. The tiny YOLO[13] network can detect persons, cars, bicycles, etc with a high confidence score (around 70 percent for humans). Next, for generating the policies for camera movements, we chose a Partially Observable Markov Decision Process (POMDP)[1] to optimize the sequential decision making problem while considering uncertainty about the noisy estimates given by the tiny YOLO[13] network in various situations.

1.2 Related Work

Object detection and recognition is performed using many advanced computer vision techniques like statistical pattern recognition and deep neural networks. There are many gait recognition techniques[2, 7, 8, 9] which aim to identify individuals against a pre-established gait database. Also, there are some techniques that identify the presence, movement and interaction of people through blob tracking[3, 5]. However, recent works have shown how POMDP[1] models can be used for intrusion detection task[11]. Zhang et al [11] used an hierarchical POMDP[1] model for solving a scene analysis problem in a robot domain. They have explicitly defined a three layered hierarchical architecture with different layers addressing certain fundamental questions like: where to look?, what to process? and finally how to process?'. However, all previous work had a drawback of performing wide range of filters, applying background subtraction and generating region of interest (ROI) before they could detect an object in a given scene that made the entire process of detection very slow. In our system, we have used the tiny YOLO[13] network that does not suffer from these limitations. We modified tiny YOLO[13] network that detects object location and the direction of movement which is then given as input to the POMDP[1] model to generate policies for camera movements.

1.3 Result Summary

Intuitively, the goal of a video surveillance system should be to move the camera towards the direction of an intruder's location. A key assumption made by our model is that the aperture of the camera corresponds to the mapped object location in the frame of reference. In our

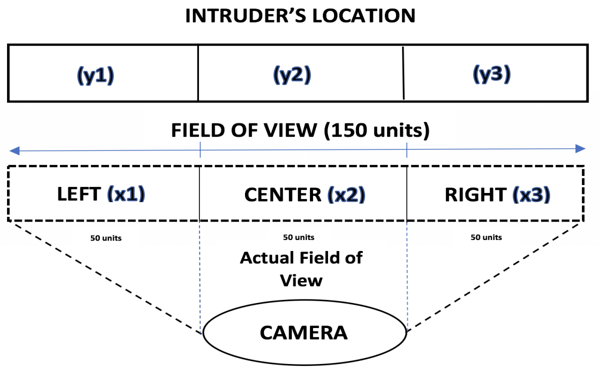


Figure 1: System and Intruder's Scenario

framework, as shown in figure 1, we divided the camera's field of view in three directions: left($x1$), center($x2$) and right($x3$) and an intruder's location as $y1, y2$ and $y3$. As the intruder moves from $y1$ to $y2$, and then $y2$ to $y3$, the POMDP[1] must generate the following policies:

Object location $x1$**, Policy** $mvx1$
Object location $x2$**, Policy** $mvx2$
Object location $x3$**, Policy** $mvx3$

In general, the policy should always have the intruder's location in camera's current field of view which we referred as object in frame. Our system successfully generated the policy that always moved the camera in the direction of an intruder.

2 Background

2.1 YOLO[13] Network

Recent approaches like R-CNN and it's various versions use region proposal methods to generate potential bounding boxes in an image. Then a classifier is run on these proposed boxes to perform post processing for refining these bounding boxes. This helps in eliminating duplicate detections. Such complex systems are slow and difficult to optimize because each individual component needs to be trained separately. But, YOLO[13] treats object detection as a single regression problem. It views the image globally and converts image pixels directly to bounding box coordinates and predicts probability of multiple classes. YOLO[13] has a very simple architecture and this makes it very efficient. The base version of YOLO[13] runs at 45 fps while the tiny YOLO[13] runs at 150 fps. Furthermore, YOLO[13] achieves more than twice the mean average precision than other real-time systems. Unlike sliding window and region proposal-based techniques, YOLO[13] sees the entire image at once and implicitly encodes contextual information about classes as well as their appearances[13].

2.2 Working

- YOLO[13] divides the input image into $S \times S$ grid where every grid cell is responsible to detect an object[13].
- Each grid cell predicts B bounding boxes and confidence scores for those boxes[13].
- Confidence Score is defined as $Pr(Object) * IOU_{pred}^{truth}$[13].
- If no object exists in that cell, the confidence score is zero. Otherwise the confidence score is equal to the intersection over union (IOU) between the predicted box and the ground truth[13].
- Every grid cell has some fixed number of bounding boxes which make 5 predictions: $x, y, w, h,$ and confidence score (fig 2). The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. w and h represent width and height respectively relative to the whole image[13].
- Each grid cell also predicts C conditional class probabilities, $Pr(Class_i|Object)$. These probabilities are conditioned on the grid cell containing an object[13].
- At test time, conditional class probabilities and the individual box confidence predictions are multiplied, $Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class) * IOU_{pred}^{truth}$, which gives class-specific confidence scores for each box[13].

2.3 POMDP[1] Formulation

POMDPs[1] are a mathematical framework for sequential decision-making under uncertainty and partial observability. Formally, a POMDP[1] is a tuple $\langle S, A, T, R, \Omega, O, \gamma \rangle$ where S, A and Ω are finite sets of states, actions and observations respectively, $T : S \times A \times S \rightarrow [0, 1]$ denotes the transition model, $R : S \times A \rightarrow \mathbb{R}$ is the reward function, $O : \Omega \times A \times S \rightarrow [0, 1]$ represents the observation model, and γ is the discount factor[11].

3 Technical Description

3.1 Our POMDP[1] Model

- $S \in CD \times OL \times OF \times OD$
  - $Camera\ State : CD \in \{x1, x2, x3\}$
  - $Object\ Location : OL \in \{y1, y2, y3\}$
  - $Object\ in\ Frame : OF \in [True, False]$
  - $Object\ Direction : OD \in \{F, B, S\}$
    - $'F' : Forward$
    - $'B' : Backward$
    - $'S' : Still$
- $A(Actions) : mvx1, mvx2, mvx3$
  - $Action\ Semantics :$
    - $'mvx1' : move\, camera\, to\, x1\, if\, object\, moves\, to\, y1$
    - $'mvx2' : move\, camera\, to\, x2\, if\, object\, moves\, to\, y2$
    - $'mvx3' : move\, camera\, to\, x3\, if\, object\, moves\, to\, y3$
- T, a stochastic transition function, i.e., $T(s, a, s') = Pr(s_{t+1} = s_2 | s_t = s_1, a_t = a)$ – the probability of executing action a from state s_1 at time t and reaching state s_2 at time $t + 1$. These can be specified by a dynamic Bayesian Network (DBN) over the state variables with the conditional probability tables (CPTs) represented by

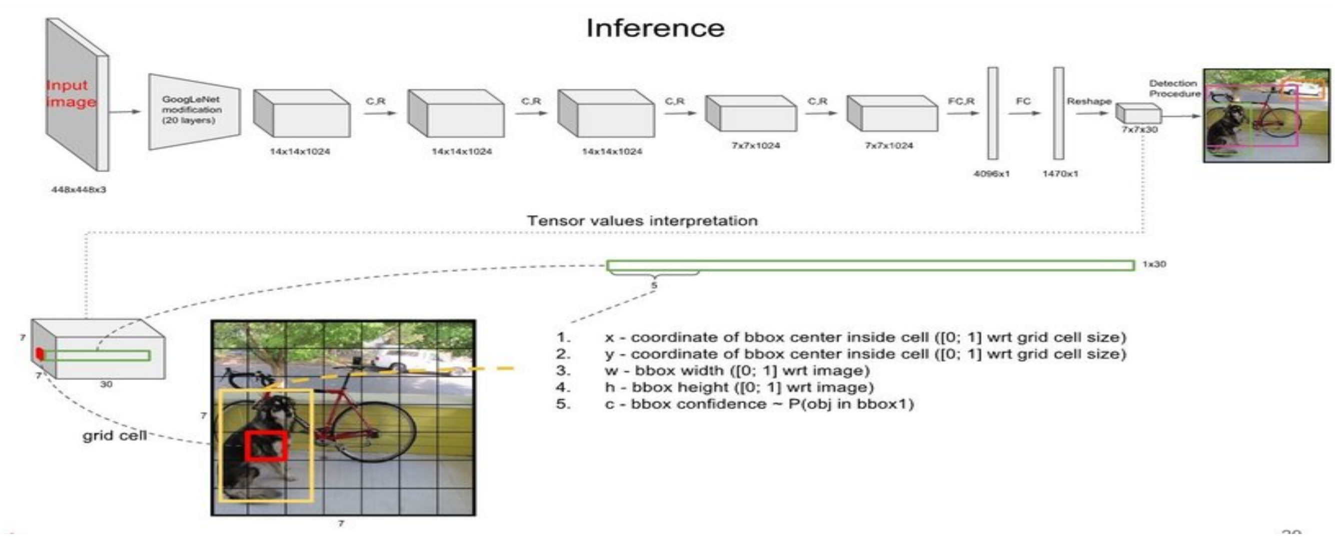

Inference
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
1x30
5
7
7
30
grid cell
7
7
1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox confidence ~ P(obj in bbox1)

Figure 2: YOLO[13] Network Architecture
[13]

algebraic decision diagrams (ADDs). The transitions exhibit considerable context specific independence, leading to a relatively compact specification of the system dynamics[11][10].

- R, reward function is defined in the table below:

Table 1: Reward Function

Camera State	OL	OF	Reward
x1	y1	Yes	+1000
x2	y2	Yes	+1000
x3	y3	Yes	+1000

- Ω, the noisy estimates of the intruder's location and direction provided by tiny YOLO[13] to the POMDP[1] model. We have assumed that camera is at a fixed location and is monitoring a specific target area.
- O, observation function that provides noisy estimates of an intruder's location and direction.

3.2 Dynamic Bayesian Network (DBN)

Real-world POMDPs[1] tend to have very large state spaces and it becomes increasingly important to exploit the structure of our problem to mitigate the complexity of state and belief spaces. Therefore, we have used a dynamic Bayesian Network (DBN) to compactly represent the transition and observation functions of a POMDP[1] and exploit conditional independence with respect to our DBN model[12].

Conditional independence refers to the fact that some variables are probabilistically independent of each other when the values of other variables are held fixed. In this graph (fig. 3), the nodes are state, action and observation variables and the edges represent probabilistic dependencies. The nodes are arranged in slices corresponding to two successive time-steps and each state variable occurs in each slice. Each node X_i in the second slice has a conditional probability table (CPT) that specifies the conditional probability distribution $P(X_i|parents(X_i))$ of X_i with respect to its parent variables. For example, as shown in figure 3, $Object\ in\ Frame$ has two parents: $Camera\ State$ and $Object\ Location$. So, the CPT will be $P(Object\ in\ Frame|Camera\ State, Object\ Location)$. Using Bayes theorem, the transition function, which corresponds to $P(X_1, X_2|X_1, X_2, A_1, A_2)$, can be factored into a product of smaller conditional distributions[4][12].

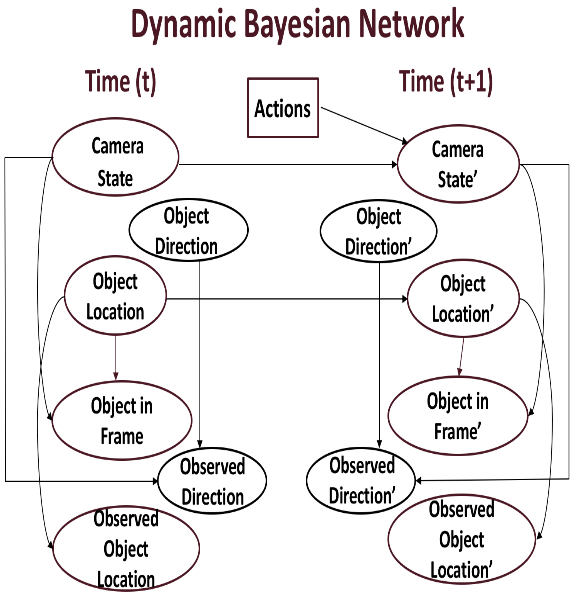


Figure 3: dynamic Bayesian Network for our POMDP Model

3.3 System Overview

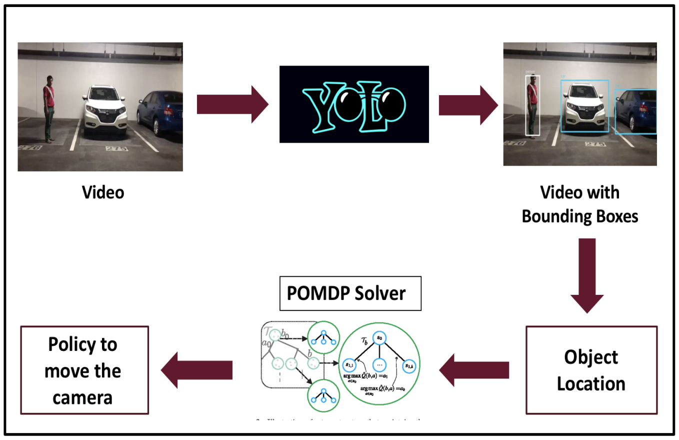
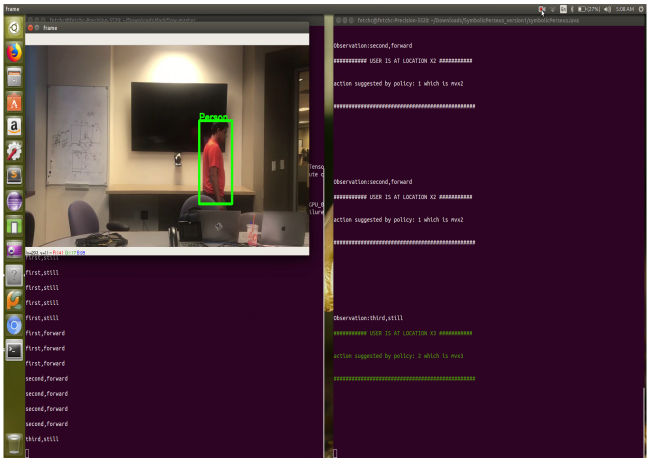


Figure 4: Smart Video Surveillance System Overview

The system overview of Smart Video Surveillance framework:

1. Camera fixed at a location is monitoring a specific target area.
2. Tiny YOLO[13] runs in this real-time setting and returns noisy estimates of intruder's location and direction.
3. Using these noisy estimates, the POMDP solver[10] generates a policy which specifies the direction in which the camera has to move to track the intruder.

4 Results



```
(x=293, y=1) ~ R:141 G:117 B:95
first,still
first,still
first,still
first,still
first,still
first,forward
first,forward
first,forward
second,forward
second,forward
second,forward
second,forward
third,still
```

```
fetchc@fetchc-Precision-5520: ~/Downloads/SymbolicPerseus_version1/symbolicPerseusJava

Observation:second,forward
########### USER IS AT LOCATION X2 ###########

action suggested by policy: 1 which is mvx2

##############################################

Observation:second,forward
########### USER IS AT LOCATION X2 ###########

action suggested by policy: 1 which is mvx2

##############################################

Observation:third,still
########### USER IS AT LOCATION X3 ###########

action suggested by policy: 2 which is mvx3

##############################################
```

Figure 5: Smart Video Surveillance Execution

We took a live video using mobile camera from a fixed viewpoint and our smart video surveillance system generated the policy as per the movement of the intruder.(Refer Table 2)

Table 2: Results

Time(sec)	Noisy Estimates	Policy Generated
1	first,forward	mvx1
10	second,forward	mvx2
15	third,forward	mvx3
20	third,backward	mvx3
25	second,backward	mvx2
30	second,forward	mvx2

5 Conclusion

In this project, we have successfully developed an intrusion detection system that integrates YOLO[13] with a POMDP solver[10] for tracking an intruder. We demonstrated the performance of our system on a live video, captured by a mobile camera. The results show that our framework is able to generate correct policies for the camera. Our system is very flexible as we can easily replace the object detection module (tiny YOLO[13]) with any state-of-the-art object detection and tracking computer vision techniques.

6 Future Work

Currently, our system controls one camera and keeps focus on one object for determining the camera policy. One possible future work could be to extend our current framework to multiple objects and cameras, demanding extensive coordination. Also, we are taking observations with minimum time lags and in real life situations, such an idealistic setting is hard to account for.

7 Individual Role

1. **Chirav Dave, Natasha Mittal, Diptanshu Purwar:** Literature Survey of Deep CNNs for Object Detection and Recognition.
2. **Chirav Dave, Natasha Mittal, Diptanshu Purwar:** Literature Survey of POMDP[1] as a framework to model stochastic dynamic policies.
3. **Chirav Dave, Natasha Mittal, Diptanshu Purwar:** For gaining a better understanding of how DBN is represented using an algebraic decision diagram, several models were created with tweaks in state and observation variables by each team member. This led to better insights in our current model.
4. **Diptanshu Purwar:** Developed the POMDP[1] model to represent the relevant features of video surveillance using a dynamic Bayesian Network.
5. **Chirav Dave, Natasha Mittal:** Played around with tiny YOLO[13] architecture and executed the same on various sample videos to give noisy estimates of intruder's location and direction.
6. **Chirav Dave, Natasha Mittal, Diptanshu Purwar:** Integration of YOLO[13] Network and POMDP Solver[10].

8 Acknowledgments

We are very thankful to Prof. Siddharth Srivastava for guiding us throughout the project.

References

[1] K. J. Aström. "Optimal Control of Markov Decision Processes with Incomplete State Estimation". In: *J. Math. Anal. Appl.* 10 (1965), pp. 174–205.

[2] J. Ben-Arie et al. "Human activity recognition using multidimensional indexing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.8 (Aug. 2002), pp. 1091–1104. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2002.1023805.

[3] Aaron F. Bobick and James W. Davis. "The Recognition of Human Movement Using Temporal Templates". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 23.3 (Mar. 2001), pp. 257–267. ISSN: 0162-8828. DOI: 10.1109/34.910878. URL: http://dx.doi.org/10.1109/34.910878.

[4] Craig Boutilier and David Poole. "Computing Optimal Policies for Partially Observable Decision Processes Using Compact Representations". In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*. AAAI'96. Portland, Oregon: AAAI Press, 1996, pp. 1168–1175. ISBN: 0-262-51091-X. URL: http://dl.acm.org/citation.cfm?id=1864519.1864560.

[5] M. Brand, N. Oliver, and A. Pentland. "Coupled Hidden Markov Models for Complex Action Recognition". In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. CVPR '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 994–. ISBN: 0-8186-7822-4. URL: http://dl.acm.org/citation.cfm?id=794189.794420.

[6] Anthony R. Cassandra. *Optimal Policies for Partially Observable Markov Decision Processes*. Tech. rep. Providence, RI, USA, 1994.

[7] Naresh Cuntoor, Amit Kale, and Rama Chellappa. "Combining Multiple Evidences for Gait Recognition". In: *Proc. ICASSP*. 2003, pp. 6–10.

[8] Quentin Delamarre and Olivier Faugeras. *3D Articulated Models and Multi-View Tracking with Physical Forces*.

[9] D. M. Gavrila. "The Visual Analysis of Human Movement: A Survey". In: *Computer Vision and Image Understanding* 73 (1999), pp. 82–98.

[10] Jesse Hoey and Pascal Poupart. *Solving POMDPs with Continuous or Large Discrete Observation Spaces*. Edinburgh, Scotland, 2005.

[11] Komal Kapoor et al. "Using POMDPs to Control an Accuracy-Processing Time Trade-Off in Video Surveillance". In: *Proceedings of the Twenty-Fourth Conference on Innovative Applications of Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. 2012. URL: http://www.aaai.org/ocs/index.php/IAAI/IAAI-12/paper/view/4788.

[12] Pascal Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. 2005.

[13] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[14] Guy Shani, Joelle Pineau, and Robert Kaplow. "A Survey of Point-based POMDP Solvers". In: *Autonomous Agents and Multi-Agent Systems* 27.1 (July 2013), pp. 1–51. ISSN: 1387-2532. DOI: 10.1007/s10458-012-9200-2. URL: http://dx.doi.org/10.1007/s10458-012-9200-2.

[13] [1] [10] [11] [14] [6] [4] [2] [7] [8] [9] [3] [5]