DESIGN OF AN AUTONOMOUS ROBOT SECURITY SYSTEM USING NEURAL NETWORKS

R. A. Balsara, S. S. Sardar and A. J. Jain

B. Tech Students, Department of Mechanical Engineering,
Vishwakarma Institute of Technology (Autonomous Institute affiliated to University of Pune)
Bibwewadi, Pune, India

ABSTRACT

In the future, intelligent machines will replace or enhance human capabilities. Artificial intelligence is the intelligence exhibited by robots and it is having a huge impact on various fields as expert systems are widely used these days to solve the complex problems in various areas as engineering, science, medicine, business, manufacturing, transport, gaming. This paper focuses on an application of Artificial intelligence in the field of security. It is based on a project that was implemented to enhance the security of a Datacenter. Security is the primary concern of modern day. Datacenters are prone to problems like fire, cyber theft, water leakage from cooling systems or presence of rodents. The system consists of a robot, periodically patrolling the datacenter and continuously looking out for these identified problems via an on-board camera module, the problems are further classified using a trained neural net. A map of the workspace in the form of a binary occupancy grid is given to the robot so that it can position and maneuver itself in the datacenter autonomously using mobile robot algorithm: Monte Carlo Localization with particle filters. Also a variety of patrolling paths can be defined by the operator. If anything is detected a vocal message is given to alert the staff and after every patrolling cycle is complete, a summary report is generated. The operator can simply examine the report and know the situation without having to monitor the video feed.

Key word: Artificial Intelligence, Security, Datacenter, Neural Networks, Monte Carlo Localization, Particle Filters.

http://www.iaeme.com/ijcet/issues.asp?JType=IJCET&VType=8&IType=3
1. INTRODUCTION
The applications and demand of robots or intelligent systems is proliferating every year, from manufacturing to healthcare to security and further. Artificial intelligence (AI) is the study of how to make computers do things which at the moment, people do better [1]. This sudden demand is primarily due to exponential shrinkage of computers in size and conversely increase in processing capabilities, as we see in this paper such a computer (i.e. Raspberry pi) is at the center of our project.

Security is a broad field and developing an autonomous system gives rise to many challenges and this paper tries to address most these issues by providing a real-time efficient solution.

To begin with, the project is divided into two major parts, the first being Anomaly Classification which is done by extracting essential features from raw images and then further classify them into various anomalies by using an artificial neural network and the second being Robot Localization which locates the robot in the given environment for optimal navigation in the given space.

Finally, datacenters are chosen as their optimal to test our approach as they provide a human scarce environment in order to minimize the random events occurring due to human interaction which might cause flawed results.

2. SYSTEM DESIGN
The system consists of a robot which periodically patrols the datacenter. It is equipped with an on-board camera module to gather live images with a frame rate of 2 fps. These images are then processed and classified. The robot maneuvers in the datacenter autonomously using the Monte Carlo Localization and Control Algorithm. Variable patrolling paths can be defined by the operator and after every patrolling cycle is complete the system generates a summary report.

2.1. Anomaly Classification
The primary objective of the system is to classify the situation in one of the five categories: fire, water, human, rodent, normal. Classification was done on the basis of features that are extracted from the images, and for this purpose firstly a neural net was trained using features extracted from labelled-image sets.

2.1.1. Feature Extraction and Clustering
A data set consisting of 671 labelled images per class was created which was used for training the neural net. SURF (speeded up robust features) were extracted from these images using MATLAB’s Computer Vision and Image Processing toolbox. 12883200 features were extracted in total, K Means Clustering Algorithm was implemented to select the prominent 500 features from these which could be efficiently used for classification.

2.1.2. Classification using Neural Network
A two-layer Feed-Forward Neural Network was trained using the labelled image data set. The data set was divided randomly into three subsets:

- Training Set consisted of 70 percent images.
- Validation set consisted of 15 percent images.
- Test set consisted of 15 percent images.
The training set is used for computing the gradient and updating the network weights and biases [9]. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial training, however when the network begins to over-fit the data, the validation error increases [9]. The network weights and bias are saved at the minimum validation set error. The test set is completely independent and is used for network generalization [9].

As shown in Figure 1, the overall Confusion matrix shows the network has an accuracy of 99.9 percent.

![Confusion Matrices](image)

**Figure 1** Confusion Matrices

The multilayer network consists of a hidden layer and an output layer. Sigmoid Transfer function is used in the hidden layer consisting of 15 neurons, similarly Softmax transfer function which is used in output layer consists of 5 neurons. Input to the neural network is an image feature vector (500 x 1) and the output is a probability vector (5 x 1).

![ Implemented Neural Network](image)

**Figure 2** Implemented Neural Network
2.2. Robot Localization

Robot Localization consists of an array of different algorithms that can be used by a robot for navigating in a pre-defined space. It is a probabilistic technique used to determine the certainty in the position of the robot at a particular position. We have implemented an instance of Monte Carlo Localization using particle filters due its high efficiency and compatibility with range-bearing sensors.

2.2.1. Monte Carlo Localization (MCL) using Particle Filters

In Monte Carlo Localization, the state of the robot is estimated by first creating random samples in the map. Each sample or particle has a weight corresponding to its range sensor measurement i.e. the higher the weight if the particle is closer to the true position and as the robot starts moving the particle gets resampled according to their weights and eventually the particles far away from the true position are eliminated, leaving behind the ones closer to the true position. [6]

Additionally, a map is given to the robot in the form of an occupancy grid. An occupancy grid is a two-dimensional logical matrix. Each cell has a value of either ‘1’ or ‘0’. ‘1’ means the grid is occupied and ‘0’ indicates it is unoccupied. The robot is set with an initial belief as the prior and calculates the posterior using the below probabilistic graphical model.

![Probabilistic model for estimating position vector](image)

The model is also known as the Markov model, where

- $u_t$ is the control input at time $t$
- $x_t$ is the position at time $t$
- $z_t$ is the sensor measurement at time $t$
2.2.2. MCL Algorithm

Algorithm MCL($X_{t-1}, u_t, z_t$):

\begin{align*}
\tilde{X}_t = X_t &= \emptyset \\
&\text{for } m = 1 \text{ to } M:\n\quad x_t^{[m]} = \text{motion\_update}(u_t, x_t^{[m-1]}) \\
\quad w_t^{[m]} = \text{sensor\_update}(z_t, x_t^{[m]}) \\
\quad \tilde{X}_t = \tilde{X}_t + (x_t^{[m]}, w_t^{[m]}) \\
\text{endfor} \\
&\text{for } m = 1 \text{ to } M:\n\quad \text{draw } x_t^{[m]} \text{ from } \tilde{X}_t \text{ with probability } \propto w_t^{[m]} \\
\quad X_t = X_t + x_t^{[m]} \\
\text{endfor} \\
&\text{return } X_t
\end{align*}

Figure 4 MCL Algorithm [5]

MCL is a recursive Bayes filter that estimates the posterior distribution of a sample or particle positions using sensor information. The algorithm consists of two steps. In the prediction step, a motion model to predict the possible location of every particle within space based on previous samples and its movement. In the filtering step, a filtering mechanism is applied to eliminate those predicted locations which are inconsistent with the current sensor information [3].

As an illustration, using Figure 6, we begin with initializing the robot with an initial belief in the map (representing the datacenter) and as the robot starts to move the particles start to concentrate towards the true position, which approximately takes about 50 to 60 centimeters.

Figure 6 Illustration of MCL
3. IMPLEMENTATION

The differential drive robot constituting of a Raspberry Pi 2B, which is the brain of the robot responsible for all the computations, a Raspberry Pi Camera module, capturing live images at 2 frames per second and 5 range-bearing sensors detecting distance and bearing (i.e. angle at which the obstacle is w.r.t to robot) of the nearest obstacle.

Furthermore, using MATLAB support package for Raspberry pi, a communication via internet protocol is established between the two. This helps to visualize and store the data sent by the Pi.

3.1. Flow Chart

![Figure 5 Flow Chart](image)

4. RESULTS AND CONCLUSION

Finally, the system was tested on a scaled model of a datacenter of which a map is given to the robot in the form of an occupancy grid. Sample props were used to simulate the anomalies. The robot was set in motion by giving a defined path to patrol and as illustrated in Figure 7, the robot successfully detects and identifies the various anomalies, the window also prints the percentage probability of each scenario being present (i.e. percentage probability of fire is 0.00, water: 0.00, human: 0.26, rodent: 99.74, normal: 0.00. Maximum being rodent so the situation is classified as rodent being present, similar is the case with Figure 8) with staggering 99 percent accuracy which indeed was predicted by the confusion matrix, shown in Figure 1.

http://www.iaeme.com/IJCET/index.asp 73 editor@iaeme.com
In conclusion, the system designed was efficiently monitoring the datacenter and accurately detecting the anomalies it was trained for initially. The system also provides dynamic sensing of the arena which static sensors or monitoring cameras fail to provide. It can at-least aid and at-max even replace humans for security purpose. Though this system was developed and trained specifically for enhancing security of datacenter against some identified problems, this system can easily be deployed in various other scenarios to serve a variety of purposes.

REFERENCES


Design of an Autonomous Robot Security System Using Neural Networks


