SIMULATION OF SENSOR RESPONSES
OF ADVANCED SECURITY SYSTEMS

by

JANAKIRAM NATARAJAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2006

ACKNOWLEDGEMENTS

First and foremost I would like to express my sincere gratitude to Dr. Lawrence Holder my mentor and advisor. His incredible patience, expertise and support paved the way to successfully complete my research work and served as a constant source of inspiration throughout my graduate study. I am greatly honoured being his student. I would like to extend my appreciation to Dr. Lynn Peterson and Dr. Manfred Huber for their participation and insight in my defense committee.

I am very greatful to my friends Anand, Avinash, Chaitanya, Jkumar, Kalai, Praveen, Priya and Vijay for all the love and support they offered me thoughout my graduate life. Special thanks to all my friends without whom my life would not have been the same.

I feel a deep sense of gratitude for my parents: Natarajan and Kousalya who carved out every aspect of my career and taught me the good things that really matter in life. Their unlimited love and support still provides a persistent inspiration for my journey in this life. I am also greatful for my elder brother Siva for rendering me the sense and the value of brotherhood. I am very glad to be a part of this family.

Lastly I thank God for always being with me, providing me with all the good in life.

April 21, 2006

ABSTRACT

SIMULATION OF SENSOR RESPONSES
OF ADVANCED SECURITY SYSTEMS

Publication No. ______

JANAKIRAM NATARAJAN, MS
The University of Texas at Arlington, 2006

Supervising Professor: Lawrence B. Holder

Security systems are becoming an increasingly important area of research. Advanced security detection and surveillance systems that integrates a variety of detection mechanisms, like signals from different kinds of sensors, is expected to yield more accurate assessment than any one sensor analyzed individually. Designing and investigating these systems, to date, has relied primarily on physical deployments and experimentation. While the quality of the results from such efforts is excellent, the need to work with the physical systems directly imposes a substantial research impediment. One obvious possibility for widening the scope of what can be investigated is to employ simulation as an alternative to experimentation with deployed systems.

Our goal is to develop a simulator for simulating infrared, millimeter wave and metal detector sensor systems. The simulator was developed using the Java 2D API. The data obtained from the simulator and the real systems were processed using the WEKA library of machine learning tools to produce threat classifiers which in turn were to be compared in order to establish the accuracies of the simulator with respect to the real system. We used t-test to compare the classification accuracies obtained using the real and simulation data. The P-value obtained from the t-test showed that the differences between the two distributions could be due to chance only. We also found that the simulated data helps in increasing the classification accuracy of the threat classifiers when it is combined with the real data. Also the agreement between threat classifiers obtained using simulated and combined data validated the accuracy of our simulator. We believe that the simulator can serve as a cost effective alternate tool for studying the characteristics of the security systems and help in constructing better threat classifiers.

TABLE OF CONTENTS

Appendix

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Problem Description

Security systems are becoming an increasingly important area of research. Advanced security detection and surveillance systems that integrate a variety of detection mechanisms, like signals from different kinds of sensors, is expected to yield more accurate assessment than any one sensor analyzed individually. Designing and investigating these systems, to date, has relied primarily on physical deployments and experimentation. While the quality of the results from such efforts is excellent, the need to work with the physical systems directly imposes a substantial research impediment. One obvious possibility for widening the scope of what can be investigated is to employ simulation as an alternative to experimentation with deployed systems.

This research work is about the simulator developed for simulating infrared, millimeter wave and metal detector sensor systems. The data obtained from the simulator and the real system were processed using WEKA, a machine learning tool and the learned classifiers were then compared to establish the accuracy of the simulator. We believe the simulator will provide a useful tool for all future work in security systems.

In chapter 2 we give a brief introduction to security systems, smart sensor networks and the recent advances in this field. In chapter 3 we discuss advanced security systems, including concealed weapon detection systems and active and passive imaging systems. We discuss the different kinds of systems, their characteristics and implementation. In chapter 4 we discuss the design of the simulator. We also discuss about Java 2D API used to develop the simulator. We explain the simulation algorithms used for simulating the sensor responses. We will also discuss the working of all sensor simulation systems.

In chapter 5 we describe WEKA, which an extensive library of machine learning algorithms. We explain briefly about the various classification algorithms used and also about the parameters considered for classification. We also explain the algorithm used to convert the sensor outputs to WEKA input format. In chapter 6 we discuss the classification results and provide a comparative analysis with respect to the real system. In chapter 7 we discuss the conclusions of the research work and directions for future work.

Finally in appendix we discuss about the specifications of the security portal being developed for the SENTRY project. We also discuss the millimeter wave and the metal detector systems used in our research project. We describe the user manual for operating the simulator and its graphical user interface.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter we give a brief introduction to security systems. We then discuss smart sensor networks and methods used to process information obtained from these networks. The recent developments in these fields are also discussed later in this chapter.

2.1 Security Systems

Security detection and surveillance is becoming an increasingly important area of research. Commercial applications such as surveillance of airports and office buildings, as well as military applications, such as monitoring the battlefield to automatically collecting strategic information, have motivated interest. For this purpose a number of visual surveillance systems and security portals are being developed. Some of the conventional systems currently used at high-security checkpoints include metal detectors for personnel and X-ray systems for hand-carried items.

But most of the conventional systems are primarily oriented only for a specific purpose and they have several of limitations. These limitations with the Conventional visual surveillance systems and security portals make them less than ideal for many applications. For instance, many airports have surveillance cameras installed to monitor security breaches. But recording the surveillance video on tapes can provide evidence only after a security breach has occurred. The alternative of dedicating a security worker to watch the live video is expensive and prone to human error. Also the existing security portals cannot detect all types of threats. Most of the security portals are used as metal detectors and they have many limitations. Metal detectors can only detect metal targets, such as ordinary handguns and knives. The sensitivity and the effectiveness of the metal detectors can vary depending on the quantity, orientation, and type of metal. Furthermore, no discrimination is possible between simple innocuous items, such as glasses, belt buckles, keys, etc., and actual threats. This leads to a rather high number of nuisance alarms. Modern threats include plastic or ceramic handguns and knives, as well as extremely dangerous items such as plastic and liquid explosives. These items cannot be detected with metal detectors. Detecting threats and safely resolving potential conflicts without violating the privacy of the individuals involved would be an important contribution to the public safety.

So more research is now done on constructing an advanced security detection system that would help in detecting a wide range of threats. Such a system is expected to yield more accurate assessment than the currently available systems.

2.2 Multiple Sensor Fusion

The basic idea here is to do a distributed detection. The system will have a number of independent sensors and each will make a local decision depending on the type of the sensor. These decisions will be mostly in binary form, which will then be combined at a fusion center to generate a global decision. Figure 2.1 shown below illustrates the parallel fusion topology, which implements this processing. For this purpose either the Bayesian or the Neyman-Pearson criterion can be used.

In case of the Neyman-Pearson formulation one assumes a bound on the global probability of false alarm. Here the goal is to determine the optimum local and global decision rules that minimize the global probability of a miss or equivalently maximize the global probability of detection. There are several ways to formulate the decision rules from the observed results. For conditionally statistically independent observations at the sensors, the optimal tests at the sensors and at the fusion center, if one exists,

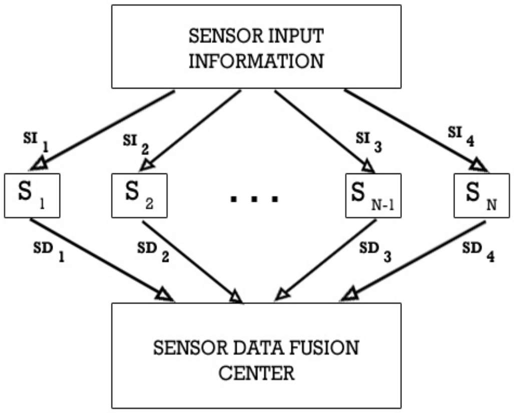


Figure 2.1. Sensor Fusion Topology.

under either of the two criteria, are likelihood ratio threshold tests [28]. The problem now becomes one of determining the optimal threshold at each sensor, as well as at the fusion center. Some methods to solve these problems were also proposed in a later work by the same author [27], which included locally optimum distributed detection of weak signals in Non-Gaussian noise under conditions of sensor-to-sensor correlated signals, nonparametric, constant false alarm rate distributed detection, robust distributed detection, and sequential distributed detection. While this task is quite non-trivial, it can still be done for a reasonably small number of sensors using iterative techniques [37]. More importantly, by using soft, multi-bit decisions at each of the sensors, it is possible to increase the performance so that it is asymptotically close to the optimal centralized scheme [36].

A tree structure can also be used to implement the distributed detection or implementation and it depends on the sensor network topology used. The work on decentralized detection of decision rules shows that the optimal decision rules are still in the form of threshold tests. Tang et al [29] considers the case where the local decisions made at a number of sensors are communicated to multiple root nodes for data fusion. With each sensor node characterized by a receiver operating curve (ROC) and assuming a Bayes risk criterion, they reformulate the problem as a nonlinear optimal control problem that can be solved numerically. Furthermore, they briefly examine communication and robustness issues for two types of tree structures: a functional hierarchy and a decentralized market. One conclusion is that if the communication costs are a primary concern, then the functional hierarchy is preferred because it leads to less network traffic. However if robustness is the primary issue, then the decentralized market structure may be a better choice.

The above methods follow a centralized topology where the information flows in one direction from the sensors to either the single fusion center or to a number of root nodes. Even in the decentralized market topology, where numerous sensors report to multiple intermediate nodes, the graph of the network is still acyclic. If the communication network is able to handle the increased load, performance can be improved through the use of decision feedback.

Pados et al. [38] examines two distributed structures:

1) A network where the fusion center provides decision feedback connections to each of the sensor nodes, and

2) A set of sensors that are fully interconnected via decision feedback.

The results show that the performance of the fully connected network is quantifiably better, but their initial system was non-robust. Robust testing functions are able to overcome this problem, and they show that robust networks tend to reject the feedback when operating with contaminated data. Alhakem and Varshney [39] study a distributed detection system with feedback and memory. That is, each sensor not only uses its present input and the previous fed-back decision from the fusion center, but it also uses its own previous inputs. They derive the optimal fusion rule and local decision rules, and they show that the probability of error in a Bayesian formulation goes to zero asymptotically. Additionally, they address the communication requirements by developing two data transmission protocols that reduce the number of messages sent among the nodes.

Swaszek and Willet [40] propose a more extensive feedback approach that they denote parleying. The basic idea is that each sensor makes an initial binary decision that is then distributed to all the other sensors. The goal is to achieve a consensus on the given hypothesis through multiple iterations. They develop two versions of the algorithm; the first is a greedy approach that achieves fast convergence at the expense of performance. The nth-root approach constrains the consensus to be optimum in that it would match that of a centralized processor having access to all the data. The main issue is the number of parleys (iterations) required to reach this consensus.

The other factors affecting the performance of smart sensors are placement of sensors, data mining and image processing techniques used to process the obtained data.

2.3 Recent Advances

In this section we discuss about some of the security systems developed recently.

2.3.1 INL Technologies

The Idaho National Engineering and Environmental Laboratory [42] through the support of national institute of justice have developed a portal that could detect concealed weapons at a much better accuracy. The device they developed was a passive device that could detect any changes in the ambient earth's magnetic field such as disturbances that are caused by metals passing through the aperture of the portal. The detector uses 16 magnetic gradiometer sensors, arrayed on both sides of the portal aperture. Data are collected from each of the gradiometers, and the change in the magnetic field over ambient background is determined. After the individual sensor responses are computed, the data from all of the sensors are processed as a group to determine the detected objects location and size.

The system uses a proprietary method to process and transform thousands of real-time data points from the portal detector array into a signature pattern for analysis. The nature of the signature produced by the object varies depending on number of factors - gait and speed of passage, proximity to center of portal and background clutter. Additionally, the orientation of the weapon can impact the signature. The signals from the sensors are first filtered from noises and other disturbances and the final output signal is then searched for suspicious patterns to detect the level of threat. The system is capable of providing a non-intrusive method for rapid detection of location and archiving of data, including visual data of potential suspects and weapon threats.

2.3.2 TSA Portal

TSA along with its industrial partners has developed a explosive detection security portal [43]. These portals are now deployed in many major airports. The security portal is known as ”puffer” machines. These machines look like conventional walk through metal detectors used in the airports. Puffs of air are blown when a person walks through the security portal and then the air is analyzed for the presence of explosives.

The transportation security laboratory is also working on another system called as the ”backscatter” [43] which uses a similar technique to detect explosives and chemicals. This machine bounce low-radiation X-rays off a person’s skin to produce images of metal, plastic and organic materials hidden under clothes. This system would produce an outline of a person walking through the portal and help in detecting explosives and metals.

2.3.3 Brijot Imaging Systems

Brijot imaging systems has developed another millimeter wave based weapon detection system [44]. The system uses passive millimeter wave for detecting threat. The system named as WDS-BDS prime is capable of detecting weapons made from a composite, ceramic or plastic (non-ferrous) material both indoor and outdoor. The system is also capable of detecting and locating a threat on a person without making them stop or entering an enclosed portal.

The system is designed to help the operators know the exact position of the threats by providing them with real-time full motion video image combined with a millimeter wave image. The system can be integrated with door locks and security alarms to enable automatic surveillance and threat detection.

2.3.4 Smiths Detection

Smiths Detection has developed an explosive and weapon detection walk-through trace portal known as Sentinel II. The Sentinel II security screening technology, was deployed at many major airports all over the united states. The Sentinel II enhances explosive detection capabilities by rapidly detecting various explosive substances that could emanate from a passengers clothing, skin or hair [44]. This portal helps to reduce the need for pat down procedures at the security checkpoints with the devices.

The system operates using the principle of Ion Mobility Spectrometry (IMS). For detecting threat the walk-through trace portal operates by passing air gently over a person from head to toe, releasing any particles that are naturally absorbed by or cling to a persons clothing or body. These particles are then drawn by a vacuum and collected for analysis. The result of this analysis is then used to detect the presence of any chemical or explosive material.

2.4 Summary

In this chapter we gave a brief introduction to security systems and smart sensor networks. Finally we discussed some of the recent advances in this field.

CHAPTER 3

ADVANCED DETECTION SYSTEMS

In this chapter we discuss the necessity and basic of advanced detection systems. We also discuss about active and passive imaging systems. Finally we discuss concealed weapon detection systems in detail.

3.1 Advanced Systems

The main objective of developing a security system is for early detection of threat with a high degree of accuracy. Hence more research is now done on constructing an advanced security detection system that integrates many individual functions of current systems and that integrates a variety of detection mechanisms, like signals from different kinds of sensors. Such a system is expected to yield more accurate assessment than any one of the sensors analyzed individually.

These advanced detection systems perform multi-sensor data fusion, where each sensor is used for a certain detection purpose. Some of the sensors that are going to be used for this purpose are visible image type sensor, which utilize images from visible wavelength cameras, infrared image sensors, which utilize images from infrared cameras, millimeter-wave sensors, which utilize millimeter waves, laser type sensors, biometric sensors and RFID tags. Previous work on these sensors did not integrate them into the same security system. Each of them was used separately and for different purposes.

Hence the basic idea here is to have a variety of independent sensors each make a local decision and then to combine these decisions at a local center to generate a global decision. Toward this goal several novel image/signal processing algorithms need to be developed which include algorithms for image registration, image enhancement, image fusion, denoising, and object extraction.

3.2 Imaging Systems

Imaging systems play a major role in today's security systems. Thermal imaging systems, millimeter wave imaging systems, low light CCD cameras and video surveillance systems are widely used for ensuring security and safety.

These imaging systems can be either active or passive. Active imaging systems transmit short bursts or 'pulses' of electromagnetic radiation in the direction of the subject. It then records the origin and strength of the backscatter received from objects within the system's field of view. On the other hand passive imaging systems sense low level electromagnetic radiation given off by all objects in the natural environment.

3.2.1 Active Imaging Systems

The active imaging systems emit their own electromagnetic energy waves. For example the active millimeter wave imaging systems emit energy at millimeter wave frequencies, and then capture the response from the target. This response is then used to get a image of the target. This technology is used in the case of radars. But the disadvantage with an active imaging system is that the target is subjected to possibly harmful electromagnetic radiation. This may cause health problems or even affect the materials. Hence this is problematic for security purposes.

3.2.2 Passive Imaging Systems

Passive imaging systems work by capturing the natural energy emitted by the subjects. All objects emit waves of the microwave frequency. The passive imaging systems are designed to capture these energy waves and then form an image of the subject. The millimeter wave sensor system and the infrared wave sensor system used in our research work are passive imaging systems. Since they don't emit any waves they can be used for security purposes for monitoring humans.

Some of the widely used active and passive imaging systems are shown in Table 3.1.

Table 3.1. Summary of Imaging Sensors

Description	Illumination	Proximity
Magnetic Imaging Portal	Active	Near
MRI Body Cavity Imager	Active	Near
Microwave Holographic Imager	Active	Near
Microwave Dielectrometer Imager	Active	Near
X-Ray Imager	Active	Near
Microwave Radar Imager	Active	Far
Broadband/Noise Pulse Millimeter-Wave/Terahertz-Wave Imager	Active	Far
Millimeter Wave Radar Detector	Active	Far
infrared Imager	Passive	Far
Passive MMW System	Passive	Far
Active MMW System	Active	Far

3.2.3 Visual Image Sensor System

Visible image sensors use imagery from the visual cameras that are widely used for surveillance purposes and employs image processing technology to isolate images and determine the information. There are two typical methods of image isolation:

a) The background discrimination method, which isolates the subject by means of differences from the background image,

b) The time subtraction method, which isolates images by comparing images captured at different time intervals.

3.3 Concealed Weapon Detection Systems

Concealed weapon detection is an increasingly important area of research. Currently number of an image sensing techniques are used to perform concealed weapon detection. The imaging sensors must possess a number of properties in order to be used for concealed weapon detection. They are

1] Penetrate heavy clothing

2] Display information in real time

3] Have a long range

But there is no sensor system which would satisfy all these requirements. For example the infrared sensor system has a very poor penetration compared to the millimeter wave sensor system which has a very short range. Hence image fusion is a key technique to achieve improved concealed weapon detection. Image fusion is a process of combining complementary information from multiple sensor images to generate a single image that contains a more accurate description of the scene than any of the individual images. Some of systems used for the detection of concealed weapons are discussed here.

3.3.1 MilliMeter Wave Sensor System

Millimeter waves are high-frequency electromagnetic waves usually defined to be within the 30-300 GHz frequency band. Some of the current systems operate at lower microwave frequency bands as well. Passive millimeter wave sensors measure the apparent temperature through emitted or reflected sources from different objects. The sensor output is a function of emissivity of the object, which is shown on the MMW spectrum of the receiver. Millimeter-wave systems are nonionizing and, therefore, pose no known health hazard at moderate power levels. The output of these systems can be of very high resolution due to their short wavelength (1-10 mm).They can penetrate fog, clothing and can work in any lighting conditions. This helps in military applications for detecting guiding missiles and targets, and in civilian applications like weather forecasting and concealed weapon detection. By using a millimeter wave it is possible to penetrate through a person's clothing and detect a concealed weapon. Humans are highly emissive, meaning they emit high levels of millimeter wave energy. Humans appear "hot" or "bright" to a millimeter wave imager-radiometer. In contrast, usually metals and guns have high reflectivity and low emissivity in millimeter wave frequencies and therefore take on the energy of any highly emissive objects in their reflection path. Plastic, ceramic and composite objects are absorptive, and, in contrast to the highly emissive human, appear cold or dark.

One of the first generation of MMW sensors is the focal-plane array MMW sensor by Millitech Corporation. Latest technology developments also help in developing video sequences at a rate of 30 frames per second from millimeter wave sensor systems. One such camera is the pupil-plane array from Trex Enterprises. It is a 94-GHz radiometric pupil-plane imaging system that employs frequency scanning to achieve vertical resolution and uses an array of 32 individual wave-guide antennas for horizontal resolution.

The images obtained from a millimeter wave sensor system are very noisy. Hence in order to detect the exact size and shape of an object in the image, various image enhancement and noise suppression techniques have to be followed. Some of the wavelet transform methods are used for noise suppression and object enhancement of these images. The current state-of-the-art in millimeter-wave imaging systems includes a variety of 2D and 3D millimeter wave imaging systems.

3.3.2 Infrared Sensor System

Infrared image type sensors are used to identify objects by isolating them using their temperature differences. The characteristic features are represented in the form of paired areas of higher and lower temperatures than in the background determined from a differential image contrasted with the background image. For example when a person is carrying an object, the temperature of the object will be different when compared to the temperature of the person. This difference will be detected by the infrared imager. The underlying concept is that the temperature of the person's body will be absorbed by their clothes and then reemitted by it. As a result there are some problems with these systems. An infrared system can only be used to detect objects when the clothing is thin, tight and stationary. Usually the human subject is at a higher temperature than its background. Hence in infrared images the body portion is darker than the background. The images obtained from an infrared sensor are subjected to histogram based thresholding operations to extract features.

3.3.3 Sensor Image Fusion

By fusing the images from different sensors like millimeter wave image data and its corresponding infrared or electro-optical image, more complete information can be obtained and this information can then be utilized to facilitate concealed weapon detection. For example previous research works has shown that fusion of images from millimeter wave sensor and infrared sensor improve extraction of the concealed weapon [32]. In addition, fusion of an electro-optical image and its corresponding millimeter wave image may facilitate recognition of a concealed weapon by locating the human subject hiding the object.

3.3.4 RFID Tag

Radio frequency identification (RFID) technology boosts the efficiency and remote sensing capabilities of sensor systems in tracking objects and persons. RFID technology is widely used for number of security purposes. AXCESS international, a leading American designer of supply chain sensors has integrated radiation detectors with RFID technology to detect radiation sources. By linking the wireless sensor network to digital video cameras that are automatically triggered by alarms, the system transmits images of the suspect cargo, vehicles, or persons to the appropriate authorities.

RFID is now being used with radiation sensors to detect chemicals and radiation levels. RFTrax incorporated introduced the first battery-operated gamma radiation sensor. This cadmium zinc telluride (CZT)-based portable radiation sensor platform, also RFID-enabled, can detect low radiation levels in a container shipment. RFID tags are also widely used for baggage handling purposes.

3.4 Summary

In this chapter we discussed in detail about advanced security systems. We discussed about different kinds of imaging sensors. Then we discussed about concealed weapon detection systems such as millimeter wave system, infrared system and RFID tag.

CHAPTER 4

SIMULATOR

In this chapter we discuss in detail about the simulator. We discuss the software used to design the simulator and the simulation algorithm. The design and implementation of millimeter wave sensor, infrared sensor and metal detector sensor simulators are also discussed in this chapter.

4.1 Simulator Design

This simulator was designed to simulate the responses of the three major sensor systems used in this research project. The three sensor systems are millimeter wave sensor system, infrared sensor system and metal detector sensor system. The simulation algorithm was implemented using Java.

4.1.1 Java 2D API

Primarily the work of the simulator involved extensive processing on graphical images in order to generate millimeter wave sensor, infrared sensor and metal detector sensor images. There are many languages usually used to design such simulators. For example both MATLAB and Java are widely using for image processing applications. This simulator was designed using the Java 2D API.

The Java 2D API introduced in JDK 1.2 provides enhanced two dimensional graphics, text, and imaging capabilities for Java programs through extensions to the Abstract Windowing Toolkit (AWT). Also Java provides a rich set of image processing operators and extensive support for image compositing and alpha channel images, a set of classes to provide accurate color space definition and conversion.

Java also helps to develop richer user interfaces. These image processing capabilities of Java made it a suitable language to develop this simulator.

4.2 Working of Simulator

The simulator simulates three different kinds of outputs for a given set of input arguments. They are:

1] Millimeter wave sensor image.

2] Infrared wave sensor image.

3] Metal detector sensor values.

As the subject can carry many threat objects in many shapes and sizes, experimentation with real sensor systems using a wide variety of threat objects is .time intensive. Hence simulation of such an activity is beneficial in order to study the various characteristics of the system. Also the output of such a simulation system must be close enough to the output of the real system for an given situation. In order to meet the long list of conceivable threat objects, a database of threat objects was collected from a real threat monitoring system i.e., from the MMW/IR sensor systems and metal detector systems, and the rest of the data was created virtually. The simulator also contains images of human subjects without any objects on them. Figures 4.1, 4.2 and 4.3 show some images of objects from the MMW sensor system and also an image of a human subject without any object.

During runtime the user has to specify different combinations of objects the human subject can carry in them. The input arguments are specified as command line arguments. The function of the simulator is to overlay the image of the threat or non-threat object from the database over the base image of a person or the subject. It sews the object image electronically over the base image of the subject such that the processed image will be very similar or identical to the image produced by the real system when the subject actually carries the threat object.

Another important factor that has to be considered while simuating the sensor outputs is noise. The output of the original system always contains some noise. For example in the output of a millimeter wave sensor system any object will not always look the same. There is always some noise in the output of the sensor which makes the result of every test run to look different. After studying the output images of our original system we found that the noise is found mostly around the region at which the object is present. So in our simulator we have few predefined noise patterns to add in the output images of the millimeter wave sensor system. Since we did not have a infrared sensor camera at the time of developing the simulator we did not study the noise patterns in the output images of the infrared sensor system. While generating the output images the simulator selects one of the noise patterns in random and adds it to the output images. The simulator adds the noise in the regions where the different objects are present. Some images of the noise patterns for the millimeter wave sensor system are shown in Appendix A.

Since the subject can carry any object in any part of the body for the reason of disguise, the simulator requires the placement of the threat object to be user controlled. The simulator has a set of base images and the user can select one of them during simulation. The base image of the subject is divided into 13 equal area regions, six at the top portion of the body and seven at the bottom portion of the body. The millimeter wave sensor system has separate set of base images for the top region and bottom regions. One of the base millimeter wave and infrared sensor images showing the thirteen regions are shown in Figure 4.1, 4.2, and 4.3.

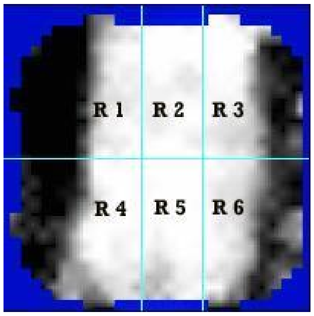


Figure 4.1. Base Upper Region of Millimeter Wave Sensor Image Showing All the Six Regions.

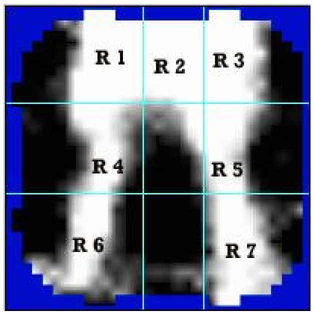


Figure 4.2. Base Lower Region of Millimeter Wave Sensor Image Showing All the Seven Regions.

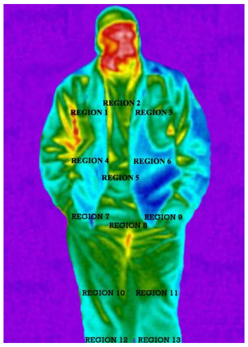


Figure 4.3. Base Infrared Sensor Image Showing All the Thirteen Regions.

The user can select any image from the database of the objects in order to place the object image over the selected base image, which will create an image of the subject with the object in the place of choice on the subject. The user has to specify the objects as command line arguments. A sample input for the simulator is explained in this section and its output images are also shown in Figures 4.4, 4.5, and 4.6.

Input:

java simulator $\langle region1\rangle$ $\langle region2\rangle$ $\langle region3\rangle$ $\langle region4\rangle$ $\langle region5\rangle$ $\langle region6\rangle$

Here the different regions correspond to the different areas in the body of the base image of a person. The regions start from the right shoulder which is region 1 and extend to the right abdomen which is region 6. An example input for the infrared sensor and millimeter wave sensor systems and their corresponding output images are shown in Figures 4.4, 4.5, and 4.6.

Input:

Java simulator: none none none Gun

Output Images: The output images are shown in Figures 4.4, 4.5, and 4.6.

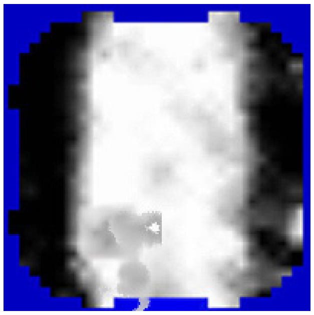


Figure 4.4. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person With a Gun in The Right Abdomen.

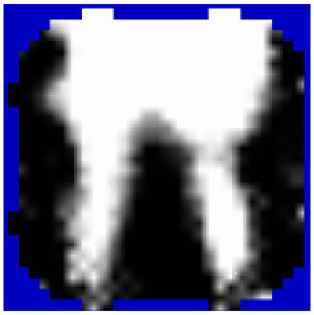


Figure 4.5. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person With a Gun in The Right Abdomen.

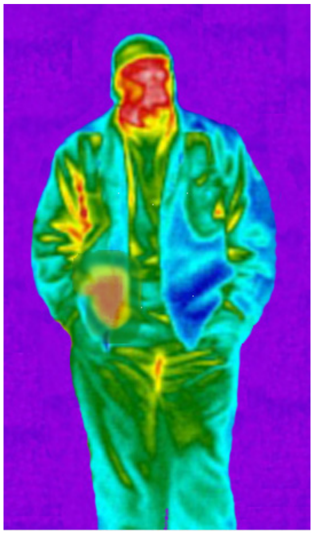


Figure 4.6. Simulated Infrared Sensor Image of a Person Showing a Gun in The Right Abdomen.

Metal detector Output:

193, 112, 75, 2, 1, 1, 96, 200, 200, 190, 196, 53, 53, 94, 64, 82, 53, 200, 200

It is also possible to simulate a person carrying multiple objects. Figure 4.7, 4.8, 4.9 and 4.10 shows examples from the millimeter wave sensor simulator. It shows a human subject carrying different objects in different regions of the body.

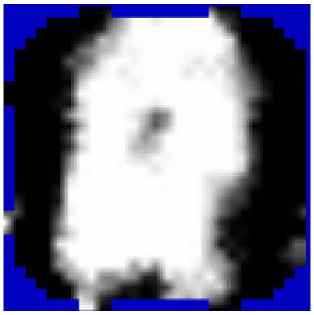


Figure 4.7. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person With a Ceramic Tile in The Center Chest.

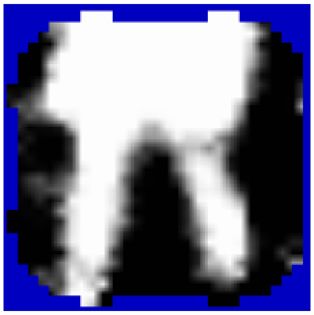


Figure 4.8. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person With a Small Tile in The Right Thigh.

The simulated images are then compared with the ones produced from the real systems. Their similarity is measured based on various machine learning algorithm's ability to learn classifiers from the simulated data that can identify threats with similar accuracy. The outputs of the simulator are analyzed using the machine learning software WEKA.

4.3 Simulation of Sensor Systems

As explained earlier the output of all three sensor systems is based on the same set of input parameters. But each sensor system has its own set of components.

4.3.1 Millimeter Wave Sensor

The millimeter wave sensor simulator uses the simulation algorithm to simulate the output images. The simulator is based on the Millivison's millimeter wave camera. The details and specifications of the real system are explained in Appendix A. The output images of the millimeter wave simulator are grayscale images of scale ranging from 0-255. The millimeter wave images of a person without any object are shown in 4.11 and 4.12.

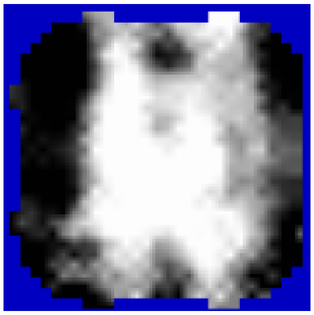


Figure 4.9. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person With a Compact Disk in The Center Chest and Iron Pipe in Center Abdomen.

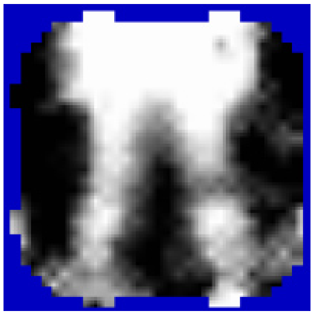


Figure 4.10. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person With a Iron Plate in Left Knee.

There are five different base images for millimeter wave system. The user can select any one of the base images for simulation.

The following list of objects can be included in the output of our simulator. The images of these objects and the base images are also shown separately in Appendix A.

1. Gun
2. Iron Handle
3. Iron Rod
4. Compact Disk
5. Metal Strip
6. Metal Plate
7. Pendant
8. Metal Barrel
9. Belt Buckle
10. Ceramic Plate

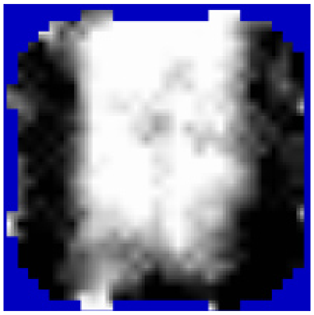


Figure 4.11. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person Without Any Object.

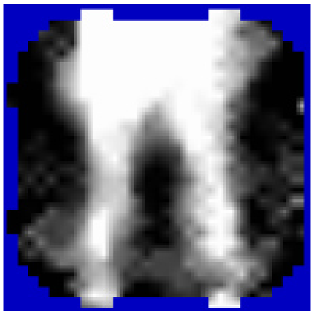


Figure 4.12. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person Without Any Object.

11. Cement Tile

12. Mobile phone

There are many other objects to be added to the database. With the above list of objects only a limited number of real life situations can be simulated. By adding more images and details into the millimeter wave sensor database many other combinations of images can be simulated. It is not possible to have a complete list of objects, because new threats will always be introduced in real life. But a fairly complete list will make this simulator a very powerful tool to analyze the real system.

4.3.2 InfraRed Sensor

The infrared sensor is another sensor system simulated by our system. As discussed earlier the infrared simulator is used to detect objects based on the heat energy emitted by them. The infrared sensor will detect the temperature difference in the subject in front of it. So if a person walks in front of the infrared sensor carrying an object, the sensor system will show the thermal image of the person with the object on his body. This is because of the temperature difference between the object and the person's body. As opposed to the millimeter wave sensor image the infrared sensor image will be a colored image. Different temperatures will be shown in different colors by the infrared sensor system.

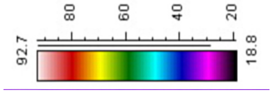


Figure 4.13. Reference Image Showing Color Difference For Different Temperatures of The Subject.

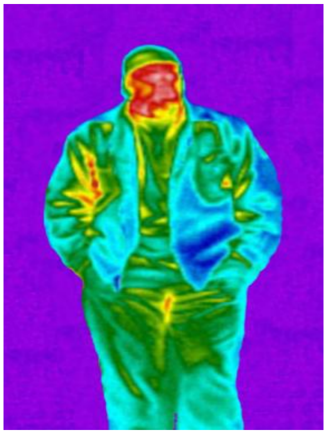


Figure 4.14. Simulated Infrared Sensor Image of a Person Without Any Object in the Body.

In order to show the temperature difference between the various regions of the subject different colors are added to the output of the infrared sensor. The reference image showing the different colors used to indicate the temperature difference is shown in Figure 4.13.

Also shown in Figure 4.14 is an image of a person taken through the infrared sensor camera. The image shows the difference of heat energy emitted at different regions of his

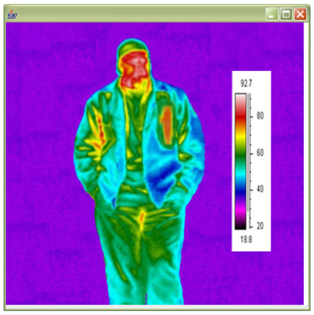


Figure 4.15. Simulated Infrared Sensor Image of a Person Showing a Metal Rod in the Left Chest.

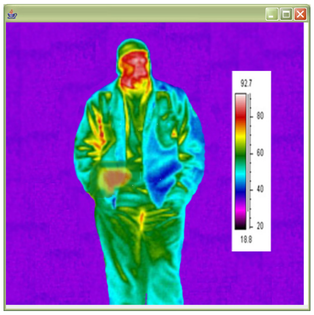


Figure 4.16. Simulated Infrared Sensor Image of a Person Showing a a Gun in the Right Abdomen.

body. The following list of objects can be included in the output of our simulator. The images of these objects and the base images are shown separately in Appendix B.

1. Gun
2. Iron Handle
3. Iron Rod
4. Compact Disk
5. Metal Strip
6. Metal Plate
7. Pendant
8. Metal Barrel
9. Belt Buckle
10. Cement Plate
11. Cement Tile
12. Mobile phone

4.3.3 Metal Detector Sensor

The simulator also simulates the values of the metal detector for a given set of input arguments. It simulates the values of a PD6500i Garrett metal detector. The specifications and details of this metal detector are explained in Appendix C. The simulated metal detector output consists of 19 different values. The magnitude of each number is generally related to the size of the object passing through a particular zone. The zone corresponds to the location of the object in the six regions of the body. This number may vary for any given input object.

The metal detector values are influenced by size of the object. It will also be influenced by the side to side location since the magnetic field, by its nature, has a gradient. The significance of the numbers is that the absence of a target produces a number of 200 and larger targets produce lower numbers.

The base values of the metal detector output are given below.

200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200

To determine detection we choose a threshold number that one or more signals must go below to set off an alarm. The value range for the first 19 fields will be between 1-200. The lower the value, the higher the amount of metal in the target.

The simulator has a database that categorizes the objects according to their size and amount of metal present in them. It also has a database that contains information about their range of influence on the various sensors of the metal detector depending upon the region of their presence. The objects are categorized into six types. Hence the influence of any object on the metal detector output depends on the category to which it belongs and according to the region it is present. Some sample examples are shown below.

Sample Input 1:

Iron Pipe in Left Chest

Output:

197, 131, 108, 088, 095, 106, 192, 200, 200, 197, 199, 054, 054, 075, 044, 084, 137, 200, 200

Sample Input 2:

Mobile in Left Chest and Compact Disk in Right Thigh

Output:

200, 183, 165, 136, 137, 142, 193, 200, 200, 198, 199, 185, 185, 195, 176, 175, 190, 200, 200

4.4 Simulation Algorithm

The basic simulator is designed completely using Java. The simulator consists of four components executing at the same time. We have developed a Graphical User Interface using Java to specify the input parameters to the simulator. The input parameters can also be specified through command prompt or an executable file having the input parameters to the simulator can also be generated. As specified in the previous section the input to the simulator is a set of images and their corresponding position in the output image. For any given set of input arguments the simulation algorithm performs image processing operations to produce the sensor outputs.

Algorithm:

```
1. Given - Set of input parameters $O_i$.
   Where $O_i \in \{Set\ of\ Images\ of\ All\ Threat\ And\ Non\ Threat\ Objects\}$.
   i $\in \{1:13\}, Set\ of\ all\ regions\ from\ 1\ to\ 13$.
     A. Check the number of input parameters N.
     B. If the number of input parameters N $\neq 13\ then\ initialize\ O_{N+1}\ to\ O_{13}\ as\ "None"$.
     C. Initialize the base values for the metal detector sensor output
```

D. Initialize the base images for the millimeter wave and infrared sensors.

E. Initialize the connection with the database containing the metal detector sensor values.

F. Define the size of the output window in terms of $(x,\ y)$, $x\ :\ Height\ and\ y\ :$ Width.

2. For every object retrieve the metal detector sensor values corresponding to the object and its region from the database. Subtract these values from the base metal detector sensor values. Print the output values of all the 19 zones of the metal detector sensor.
3. Construct image buffers for all the objects specified as input parameters and also for the base image.
4. Initialize the variables to represent all the images used to simulate the sensor outputs.
5. Allocate a separate image buffer for each image based on its dimensions.
6. Position the selected base image at the center of the output window.
7. Initialize the alpha value for the alpha compositing operations. This value is used to control the alpha value of the image pixels during blending operations. The alpha value of each pixel indicates its opacity.
8. Draw the images of the objects over the base image in the output window. The pixel intensity of the images of the objects will be controlled by the alpha value defined in the previous step. The position of the object images will be controlled based on the specified region.
9. Save the simulated output images to the specified folder. Save the metal detector output in a text file on the same folder.

4.4.1 Sample Execution of Algorithm

The execution of this algorithm is demeonstrated here,

1] Input:

Java simulator: none none Rod

2] O_1 *none* , O_2 *none* , O_3 *Rod* ,

3] N = 3,

4] O_4 *none* , O_5 *none* , O_6 *Rod* , O_7 *none* , O_8 *none* , O_9 *Rod* , O_{10} *none* , O_{11} *none* , O_{12} *Rod* , O_{13} *none* ,

5] Base metal detector values:

199, 200, 200, 200, 200, 200, 200, 200, 200, 199, 198, 200, 199, 198, 200, 200, 200, 200, 200

6] Output window size: Width = 200, Height = 200,

7] Metal detector Output:

193, 112, 75, 2, 1, 1, 96, 200, 200, 190, 196, 53, 53, 94, 64, 82, 53, 200, 200,

8] Initialize 13 image buffers for the objects and 3 image buffers for the output images,

9] Initialize the alpha value of 0.9 for compositing operations,

10] Output Images:

4.4.2 Using the Simulator

The simulator has a user interface to help the users to select their choice of base images and objects. The user can select one object for each of the 13 different regions. The working of the GUI and the user guide are explained in the appendix D.

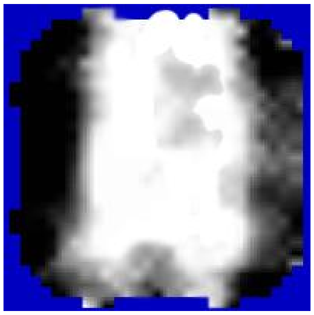


Figure 4.17. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person Showing a Metal Rod in the Left Chest.

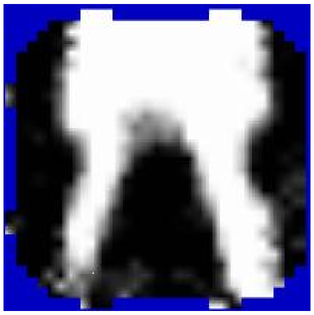


Figure 4.18. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person Showing a Metal Rod in the Left Chest.

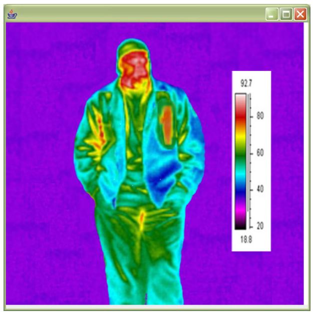


Figure 4.19. Simulated Infrared Sensor Image of a Person Showing a Metal Rod in the Left Chest.

4.5 Summary

In this chapter we discussed in detail about our simulator. The design and implementation of millimeter wave sensor, infrared sensor and metal detector sensor simulators were explained in this chapter. Finally we explained the algorithm used by the simulator and its sample implementation.

CHAPTER 5

EXPERIMENTAL SETUP

Analysis of the accuracy of the simulator involves its ability to produce results similar to the real system. For this purpose we use various classification algorithms. In this chapter we will consider the various comparison parameters involved. Also we concentrate on the machine learning algorithms used to analyze the simulated data.

5.1 Comparison Parameters

The main factor to be considered to analyze the performance of the simulator is its ability to produce results similar to the real sensor systems. In the case of our simulator it produces outputs similar to the three major sensor systems used in our project. The output of the millimeter wave sensor system and infrared sensor system are images, while the output of the metal detector sensor system is numeric. The output of the sensor systems shows whether the given situation is threat or a non-threat. Various machine learning algorithms are used to classify the outputs into one of the two categories.

We asses the statistical similarity between the outputs of the real system and the simulator using t-test. For this we first determine the classification accuracy for the real and simulator data using the algorithms. Then we perform the t-test on a set of classification accuracies obtained using the real and simulated data. Also the ability of the classification algorithms to differentiate between threat and non-threat shows the similarity between the outputs. The output of the real systems is based on different real life situations. Any situation is considered to be a threat or non-threat based on its environment, factors influencing the situation and also the security conditions.

For any given set of situations the output of the real system is first determined. Then the simulator will be used to simulate the output for similar situation. The outputs are then fed to the classification algorithms to differentiate between threat and non-threat situations. The classification accuracy of the algorithms on the two sets of outputs are used to deduce our results. Any given situation can be a threat to non-threat depending upon various conditions. In our research work we are assuming two different conditions to differentiate a threat from a non-threat.

5.1.1 One Particular Object

The presence of one particular object in the output image of the imaging sensors and a low metal detector sensor value is considered a threat. The presence of any other objects in this situation will not be considered a threat. Hence all other situations where the particular object is not present are considered to be non-threat. For example if a gun is present among the set of objects in the input range of the sensors, then the output of the sensors at that situation will be considered a threat. The presence of any other object will not make the situation a potential threat.

5.1.2 Two or More Objects

The presence of two or more specific objects in the output of the imaging sensors will be considered a threat, and the rest will be considered non-threat. For example if both iron handle and metal plate are present among the set of objects in the input range of the sensors, then the output of the sensors in that situation will be considered a threat. The presence of any other objects or even if one of them is present will not make the situation a potential threat.

5.2 T-Test

T-test is an statistical test to determine the probability of difference between two data sets. The output of the t-test is a P-value, whch corresponds to the probability that the two distributions are different from each other. According to this test a P value less than 0.05 shows that there is a significant difference between the data sets else the difference between the distributions is caused by chance only. As we know earlier statistics will never give you a yes or no answer, but a probability for yes and for no. The bottom-line is that if P is less than 0.05 the data sets are significantly different else there is no proof that they are differnet.

In our research work we need to prove that the distributions obtained from the real and simulator data are not different from each other. We can also extend our resuls and prove that the two distributions lie within an epsilon value of each other, where epsilon is some positive integer other than zero. Hence we need a P value greater than 0.05. We used the inbuilt t-test function in Microsoft excel for testing our data.

5.3 WEKA

WEKA stands for Waikato Environment for Knowledge Analysis. WEKA is a collection of machine learning algorithms for data mining tasks. The system is written in Java. The algorithms in WEKA can either be applied directly to a dataset or called from Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

The following algorithms were used for classification purposes in this project.

Algorithm 1: Support Vector Machine

Algorithm 2: Decision Tree classifier

Algorithm 3: Neural Network classifier

The output of the simulator is used to learn the difference between a subject that is a threat and a non threat. The input to the WEKA package must be in ARFF format. In order to represent an image in ARFF format it has to be converted to a suitable n dimensional vector representation.

Each element of the vector will then be considered as a feature of the image. These feature vectors derived from different images are then used as input to the algorithms in WEKA.

5.4 Classification Algorithms

The main advantage of using WEKA is to apply the learning methods to a dataset and analyze its output to extract information about the data. These learning methods are called classifiers. Here we use the classifiers from WEKA in order to analyze the classification accuracy of our simulator data. Classification here means the problem of correctly predicting the probability that an example has a predefined class from a set of attributes describing the example. Also the learning algorithms in WEKA can be applied and then the best one can be used for prediction purposes. We use three widely used classification algorithms in our experiments. These algorithms were selected based on their relative merits in the image classification domain compared to the other algorithms [1][11][22][29].

5.4.1 Neural Network Classifier

The neural network classifier is used for many pattern recognition purposes. It uses the backpropogation algorithm to train the network. The accuracy of the neural network classifiers does not depend on the dimensionality of the training data.

5.4.2 Support Vector Machines

The support vector machine classifiers work by generating functions from the input training data. This function will then be used as a classification function. They operate by finding a hypersurface in the space of possible inputs. This hypersurface will attempt to split the positive examples from the negative examples i.e., threat from non-threat. If the dimensionality of the input data is high then the SVM takes more time for training.

5.4.3 Decision Tree Classifier

The decision tree classifier is a tree based classifier which selects a set of features and then compares the input data with them. Learned patterns are represented as a tree where nodes in the tree embody decisions based on the values of attributes and the leaves of the tree provide predictions. The main advantage of a decision tree classifier is its classification speed. WEKA uses the J48 decision tree which is an implementation of the C 4.5 algorithm.

5.5 Testing Strategies

Different testing strategies can be used to train and test based on the given datasets. For our experiments we used averaging and 10-fold cross validation testing techniques.

5.5.1 K-Fold Cross Validation

The first process of training and testing the given dataset is by using the k-fold cross validation method. In a k-fold cross-validation, the given sample data set of m instances is first divided into k subsets (folds) each of size m/k. For each fold, the classifier is allowed to train using the remaining subsets and the current subset is used for testing purposes. Hence during the process the data set has to be first divided into k subsets. Then the classification algorithms are fed with these subsets of data. The left-out subsets of the training data will be used to evaluate classification accuracy. In our experiments we used 10 fold cross validation because 10 is the default value of k in WEKA.

5.5.2 Averaging

In the averaging process the classifier takes the average of a set of runs (which are typically cross-validation runs). In the averaging process we divide the given dataset into two parts. One part is first used to train the classification algorithm. So the percentage of data to be used for training purposes should be specified first. Then concepts learned during the training process are used to test the remaining data.

5.6 Vector Representation

The input for WEKA must be in ARFF (Attribute Relation File Format) format. Hence the output images and the metal detector output must be converted to their corresponding ARFF format. ARFF is an ASCII text file that describes a list of instances sharing a set of attributes. It also describes the class to which the instances belong. Any input data is represented as a collection of its feature values. So in our research work the conversion algorithm is used to convert the sensor outputs to their corresponding feature vectors. In an ARFF file the features are represented as numeric values. An example ARFF file for a set of sensor data is shown in appendix D.

5.6.1 Conversion Algorithm

The conversion algorithm is used to convert the given images to their corresponding feature vectors. The algorithm averages the gray scale values of pixel groups in the input image. The size of a pixel group is either a user defined or a predefined value. Any input image is first divided into m equal sized n x n regions. Then the gray scale values of all the pixels in each region are averaged to get the components of the feature vector. The averaged values from all these m regions form the feature vector of the given image.

1. Given - Path of a folder containing millimeter wave, infrared sensor images, and the text file having metal detector sensor value.
2. Check the number of input parameters. If the number of input parameters is two then the second parameter is n.

   Where n is the number of image pixels to be averaged.

   A. Initialize one three-dimensional array and one one-dimensional array to carry the values of the pixels.
3. Construct an image buffer to store the input image and load the input image into the buffer for performing image processing operations.
4. Get the width and height of the input image.
5. Instantiate a PixelGrabber object to store the numeric pixel data in a one dimensional array. Invoke the grabPixels() method on the PixelGrabber object to extract the pixels from the image into the one dimensional numeric array.
6. Convert the byte pixel data in the one dimensional array to a three dimensional bitwise array in order to make it easier to work with the pixel data. Use AND and bitwise right shift operations to mask all but the correct set of eight bits. The three dimensional array will contain the red, blue, green and alpha values of the image pixels.
7. Calculate the average grayscale value of pixel regions in the image of size n x n

to obtain m different values.

The averaged set of m values represent the feature vector for the given image.

5.6.2 Sample Execution of Algorithm

1] Input:

Upper.png, Lower.png, Metal.txt and Threat.txt,

2] Ouput:

56, 79, 135, 105, 73, 31, 194, 215, 172, 18, 37, 238, 242, 222, 16, 47, 243, 249, 230, 48, 75, 184, 181, 185, 66, 59, 164, 129, 129, 69, 28, 224, 213, 220, 50, 29, 167, 39, 179, 24, 60, 169, 25, 153, 63, 74, 149, 85, 130, 76, 200, 199, 199, 188, 182, 155, 151, 200, 200, 188, 198, 185, 185, 179, 170, 186, 191, 200, 200, yes

5.7 Summary

In this chapter we discussed the various comparison parameters used to compare real and simulator data. We discussed all the conditions that would differentiate a threat from a non-threat. We gave a brief introduction to t-test and WEKA. We explained the algorithm used to convert the sensor values to corresponding feature vectors.

CHAPTER 6

EXPERIMENTAL RESULTS

In this chapter we perform a comparative analysis of the simulator output and the output of the real system. Our main aim is to analyze the performance of the simulator when compared with the real system based on the classification accuracy. We perform our tests on the datasets obtained from the sensor outputs. We also discuss the performance of the different classification algorithms in this domain. We conclude by analyzing the closeness of the simulator when compared to the real system.

6.1 Classification Data

We use data from the real sensor systems and the simulation system for classification. This dataset consists of images from millimeter wave sensor system and numeric output from metal detector sensor. Each of the images from the millimeter wave sensor is of size 200 x 200 pixels. All these images are grayscale images of scale 0-255. The values from the metal detector may vary from 0-200. This data has a high variety of threat and non-threat objects of different shapes and sizes. Different classification tests were performed using single classifiers with different sets of parameters. The sensor outputs are first converted to their corresponding feature vectors. They are then combined to form one feature vector which represents both the sensor outputs.

The output of the millimeter wave sensor system is images. These images are first fed to the conversion algorithm to convert them into feature vectors. Then the output values of the metal detector sensor are combined with this feature vector to form one large vector. The vectors are then used to form the ARFF file for WEKA.

The vectors are classified as either positive or negative. The data that are considered as a threat are classified as positives and those considered non-threat are classified as negatives. For our experiments we used 124 real and 124 simulated profiles. Each profile corresponds to a person carrying a different object or one without any object. The number 124 which corresponds to the number of real and simulated profiles does not have any significance with the experiments.

6.1.1 Real Data

The data obtained from the real system are first converted to their corresponding feature vectors. Then the algorithms are used to test the classification accuracy on this dataset. The data set or profile containing a metal rod and a metal plate are considered a threat and the rest are considered non-threat. Some of the images obtained from the real system are shown in Figure 6.1 and Figure 6.2.

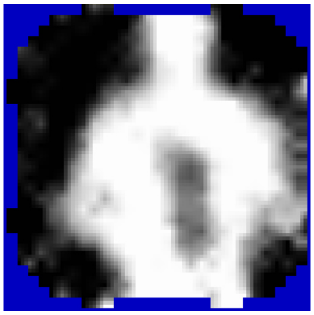


Figure 6.1. Millimeter Wave Sensor Image Showing a Man with a Large Metal Plate in His Abdomen.

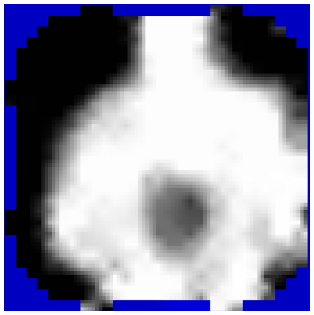


Figure 6.2. Millimeter Wave Sensor Image Showing a Man with a Compact Disk in His Abdomen.

6.1.2 Simulator Data

6.1.2.1 Experimental Case 1

For the first case we consider only one object as a threat. In our test cases we considered the presence of a gun as a threat. So if a gun is included in the input profile of the millimeter wave sensor then the output is considered a threat. Figures 6.3, 6.4 and 6.5 show the presence of a gun in the millimeter wave sensor outputs. The corresponding metal detector values are also included while constructing the feature vector for this output. For this experimental case we used the simulator data only. This experiment was performed to test the classifiers ability to learn the simulated data. Since we did not have a gun with us we did not compare the results of our simulator with the real system.

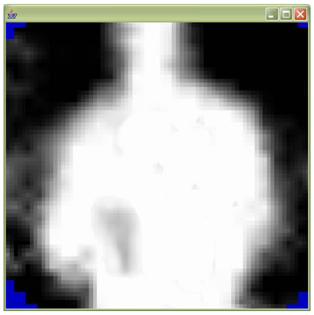


Figure 6.3. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person with a Gun in the Right Abdomen.

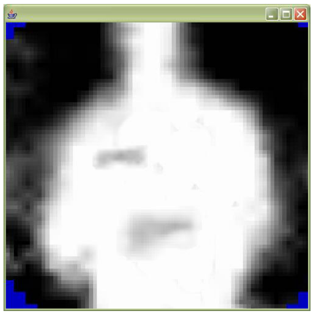


Figure 6.4. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person with a Metal Strip in the Right Chest And a Gun in the Center Abdomen.

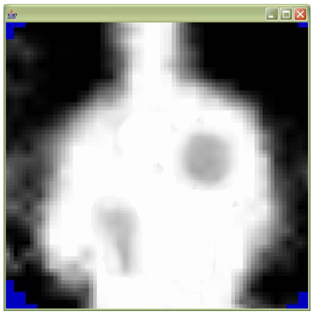


Figure 6.5. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person with a Compact Disk in the Left Chest and a Gun in the Right Abdomen.

Figures 6.3, 6.4, and 6.5 clearly show the presence of a gun in the simulated millimeter wave sensor outputs. These images are then fed to the conversion algorithm to form a feature vector and then tested using the learners'.

6.1.2.2 Experimental Case 2

For the second case we consider the presence of two or more objects of a certain kind as a threat. In our test cases we considered the presence of an iron rod and metal plate as a threat. So only if both these objects are included in the input profile of the millimeter wave sensor the output is considered a threat. So for this case we generated 124 profiles using both the real and simulator systems. Then we generated three sets of data for testing purposes. One using the real data, one using the simulated data and the third using the combination of these two data sets. For testing we use the classification accuracies obtained using these data. The Figures 6.6, 6.7, 6.8 and 6.9 show some samples of this kind.

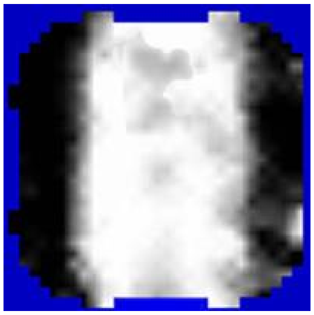


Figure 6.6. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person with an Metal Rod in Center Chest.

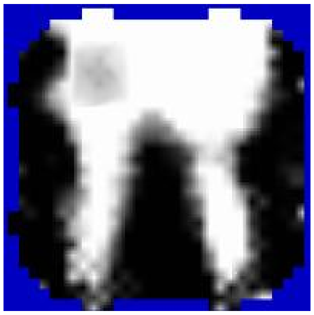


Figure 6.7. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person with an Metal Plate in Right Thigh.

6.2 Classification Results

The datasets from the real system and the simulator are tested for classification accuracy using the three algorithms discussed earlier. All the examples were classified

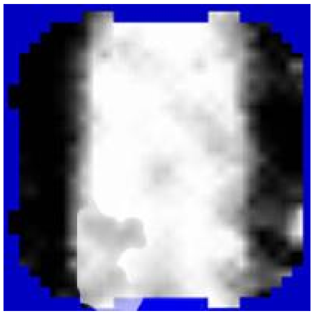


Figure 6.8. Simulated Millimeter Wave Sensor Image Showing the Upper Region of a Person with a Metal Pipe in the Right Abdomen.

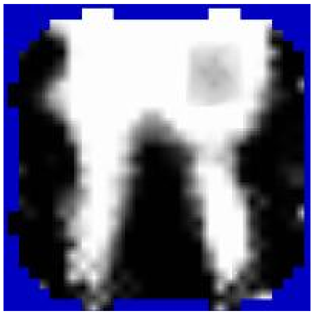


Figure 6.9. Simulated Millimeter Wave Sensor Image Showing the Lower Region of a Person with an Metal Plate in Left Thigh.

into one of the two classes, either positive or negative. For our research work since we don't have an infrared camera in our lab to verify our simulator output, we used only the millimeter wave sensor output and metal detector output for testing purposes. Here accuracy is the percentage of examples the algorithm classifies correctly. In our research work first we determined the classification accuracy using the real data and then by using the simulated data.

For determining the statistical similarity between the two data we used t-test. The t-test was performed on the classification accuracies obtained using the two data sets. Since we are using 10-fold cross validation in our experiments we determine the accuracy for each fold. Then the set of accuracies obtained using the simulator and real data are compared using t-test. We obtained a P value greater than 0.05 which showed that the difference between the two distributions could be due to chance only. We also proved that the number of misclassifications decreases if the simulator data is used along with the real data to train the learners'.

6.2.1 Real System Versus Simulator

Table 6.1 shows the results obtained using the data from the real system. The classification method used is 10-fold cross validation. The results show the accuracy obtained using 124 different datasets or profiles. For averaging the pixels of the millimeter wave sensor image we used two different values. For first set of experiments we averaged the pixels in the different regions of the image and then for the second set of experiments we averaged pixel regions of size 10 x 10. Later while discussing the accuracy obtained for simulated data we will explain the necessity for the second set of averaging data. It is also possible for the user to define their own pixel window size. Table 6.1 shows the classification accuracy and the number of misclassified instances obtained for the real data averaged based on the regions and Table 6.2 shows the results obtained by averaging pixel groups of size 10 x 10.

Table 6.1. Classification Results For Real Data by Averaging Pixel groups at Different Regions

Classification Algorithm	Classification Accuracy	Number of Misclassified Instances
Decision Tree - J 48	**91.94**	**10**
Multilayer perceptron	**97.11**	**5**
Support Vector Machine	**95.19**	**7**

From the Table 6.1 we know that the multilayer perceptron performs the best of all the three classifiers. But even though the number of misclassifications is more with the other two classifiers, the difference is not very big. We can say that all the three algorithms perform well with the real data and the parameters used in this experiment. Table 6.2 shows the classification accuracies obtained for the real data by averaging pixel groups of size 10 x 10.

Table 6.2. Classification Results For Real Data by Averaging Pixel groups of Size 10 x 10

Classification Algorithm	Classification Accuracy	Number of Misclassified Instances
Decision Tree - J 48	**90.33**	**12**
Multilayer perceptron	**95.96**	**5**
Support Vector Machine	**92.74**	**9**

From the Table 6.2 we can see that the number of misclassifications has increased. But this difference is not very high. Also later the results obtained from the combined dataset showed that the results improved when pixel groups of window size 10 x 10 were averaging. Now we will look at the accuracies obtained using the simulator data for similar kind of profiles. we used 10-fold cross validation for testing the accuracies. Like the real data we used two different values for averaging the pixel values of the simulator data. Table 6.3 shows the classification accuracy and the number of misclassfied instances obtained for the simulator data averaged based on the regions and Table 6.4 shows the results obtained by averaging pixel groups of size 10 x 10.

Table 6.3. Classification Results For Simulator Data by Averaging Pixel groups at Different Regions

Classification Algorithm	Classification Accuracy	Number of Misclassified Instances
Decision Tree - J 48	**94.35**	**7**
Multilayer perceptron	**96.78**	**4**
Support Vector Machine	**82.25**	**22**

From the Table 6.3 we know that the multilayer perceptron performs the best of all the three classifiers even for the simulator data. The performance of J 48 is also good but the performance support vector machine classifier has decreased for this dataset. Hence for increasing the support vector machine classifier's performance we have introduced the 10 x 10 pixel averaging tests in all our experiments. Table 6.4 shows the classification accuarcies obtained for the simulated data by averaging pixel groups of size 10 x 10.

Table 6.4. Classification Results For Simulator Data by Averaging Pixel groups of Size 10 x 10

Classification Algorithm	Classification Accuracy	Number of Misclassified Instances
Decision Tree - J 48	**92.74**	**9**
Multilayer perceptron	**99.19**	**1**
Support Vector Machine	**98.39**	**2**

For the Table 6.4 we can see that the performance of support vector machine classifier has increased considerably on our simulator data with 10 x 10 averaging. In order to show the usefulness of our simulator we performed another set of experiments by combining the simulator data with the real data. We named this set of data as "combined data". By comparing the number of misclassifications obtained using the combined data with that of the real data we can study the usefulness of the simulator data. For example if the number of misclassifications is less with the combined data compared to the sum of misclassifications obtained with the real and simulator data, then it shows that the learners' can learn well when the simulator data is also used along with the real data to train them. Table 6.5 shows the classification accuracy and the number of misclassified instances obtained for the combined data averaged based on the regions and Table 6.6 shows the results obtained by averaging pixel groups of size 10 x 10.

From the Table 6.5 we know that the both J 48 and multilayer perceptron performs well for the combined data. In case of the J 48 decison tree classifier the number of misclassifications has drastically decreased to 1 from 17, which is the combined value of

Table 6.5. Classification Results For Combined Data by Averaging Pixel groups at Different Regions

Classification Algorithm	Classification Accuracy	Number of Misclassified Instances
Decision Tree - J 48	**99.60**	**1**
Multilayer perceptron	**97.58**	**6**
Support Vector Machine	**86.69**	**33**

number of misclassifications with the real data, 10 and the simulator data, 7. In case of the multilayer perceptron classifier the number of misclassifications has decreased to 6 from 10, which is the combined value of number of misclassifications with the real data, 5 and the simulator data, 4. The performance of support vector machine classifier is not so convincing in this case, but we observed that its accuracy increased when pixel windows of size 10 x 10 were averaged which is shown in the Table 6.6. Table 6.6 shows the classification accuracies obtained for the combined data by averaging pixel groups of size 10 x 10.

Table 6.6. Classification Results For Combined Data by Averaging Pixel groups of Size 10 x 10

Classification Algorithm	Classification Accuracy	Number of Misclassified Instances
Decision Tree - J 48	**96.78**	**10**
Multilayer perceptron	**98.80**	**3**
Support Vector Machine	**96.78**	**10**

As we discussed earlier even the support vector machine classifier performs well with the combined data if pixel groups of size 10 x 10 are averaged. The number of misclassifications has decreased to 10 from 11, which is the combined value of number of misclassifications with the real data, 9 and the simulator data, 2. The performance of the other learners' is also good in this case compared to the previous results in Tables 6.4 and 6.2.

All the results tabulated in the Tables 6.1 to 6.6 correspond to the experimental case 2, where we are considering the presence of a metal rod and a metal plate as a threat. This is because of the availability of limited number of objects in real time. For all these experiments we compared the performance of our simulator with the real system. For the experimental case 2 we used only objects like metal plate, metal rod, compact disk, ceramic tile, cement plate and cellular phone. Now in order to test the learners' capability to learn the simulator data we test the classifiers with only the simulation data. For the experimental case 1 we don't compare our results with the results of the real data. We test the learners' capability to learn the presence of a gun in our data. Table 6.7 shows the classification accuracies obtained for the simulator data set of experimental case 1.

Table 6.7. Classification Results For Simulator Data For Experimental Case 1

Classification Algorithm	Classification Accuracy
Decision Tree - J 48	**96.97**
Multilayer perceptron	**96.97**
Support Vector Machine	**93.94**

From Table 6.7 we know that the learners' can efficiently learn the simulated data for the presence of threat objects like gun, knife and explosives.

6.2.2 T-Test

As explained earlier we used t-test to prove that the difference between the two distributions is caused by chance only.For testing the two data sets we compared the classification accuracies obtained from classifiers. Since we used 10-fold cross validation in our experiments we compared the accuracy obtained for each fold between the real and simulated data using t-test. For example we obtain the classification accuracies for all the 10 folds using one of the classifiers in the real data and then we obtain the classification accuracies for all the 10 folds using the same classifier in the simulator data. Then we compare these accuracies using t-test to obtain the P value. We tried to prove our null-hypothesis using this P value. According to our tests a P value greater than 0.05 would show that the null-hypothesis is not false.

In our experiment we averaged the values of 10 t-tests to obtain the average P value for one particular classifier. So we got three P values one for each algorithm and all of them were greater than 0.05. From this we can say that difference between the two distributions could be due to chance only and there is a certain similarity between the two distributions. The resuts of our t-tests are tabulated in Table 6.8.

Table 6.8. Average P Values Obtained By Running T-Test Over the Classification Accuracies of the Real and Simulator Data

Classification Algorithm	Average P Value
Decision Tree - J 48	**0.37**
Multilayer perceptron	**0.21**
Support Vector Machine	**0.28**

Table 6.8 shows the average P value of the t-test performed on the outputs of the three classifiers. As specified earlier a P value greater than 0.05 indicates that the two data sets are not significantly different. We can see that the P value is always higher than 0.05. For J 48 decision tree classifier we obtained a P value of 0.37, for support vector machine classifier the P value was 0.21 and for multilayer perceptron classifier the P value was 0.28.

6.3 Comparative Analysis

In the previous section we inferred that there is a certain probability that the two distributions are not significantly different. We used statistical t-tests to compare the two distributions. Table 6.8 shows the results of our t-tests. We can also extend our results to show that the distributions lie within an epsilon of each other. Our results also prove that the simulator can help in better detection of threat. In this section we analyze the results obtained from our experiments and provide an comparative analysis of the simulator and real data.

Using the results from tables 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6 we can infer that the simulator can improve the classifier's threat detection capability. This works for all the three classifiers used in our experiments. The classifiers learned better when simulator data was used along with the real data to train them. Also using the combined data we were able to reduce the number of misclassifications which shows the usefulness of our simulator.

The J 48 decision tree classifier performed very well with all the three datasets i.e., real, simulator and combined data. Figures 6.10, 6.11 and 6.12 show the learned trees for the real data, simulator data and combined data. From the figures 6.11 and 6.12 we can see that the trees learned using the simulator data and the combined data are similar. This tells us that a J 48 classifier trained by the simulator data would perform similar to the one trained by the combined data. From table 6.5 we know that there was only 1 misclassified data when the tree was trained using the combined data. So we can infer that training a tree only with the simulator data or in combination with the real data will improve the classifier's performance.

This was also true for the multilayer perceptron and support vector machine classifiers also. Tables 6.5 and 6.6 show the improvement in the classifier's performance with the combined data.

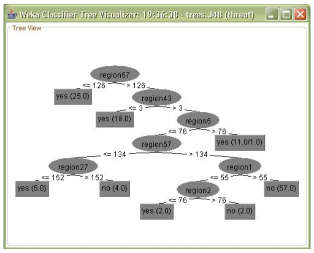

Weka Classifier Tree Visualizer: 19:36:38 - trees.J48 (threat)

Figure 6.10. Tree learned by Training the J 48 Decision Tree Classifier Using Real Data.

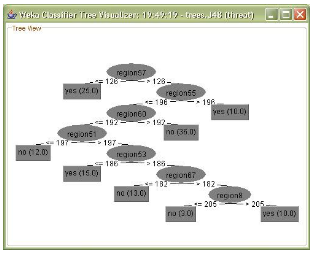

Weka Classifier Tree Visualizer: 19:49:19 - trees.J48 (threat)

Figure 6.11. Tree learned by Training the J 48 Decision Tree Classifier Using Simulated Data.

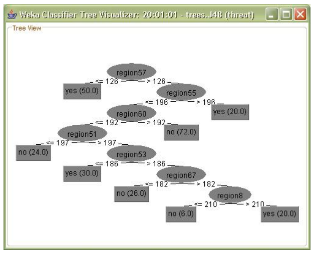

Weka Classifier Tree Visualizer: 20:01:01 - trees.J48 (threat)

Figure 6.12. Tree learned by Training the J 48 Decision Tree Classifier Using Combined Data.

The results we obtained for the t-test and the using the combined data show that our simulator could produce results similar to that of the real system. The results obtained from the combined data show that the simulator can improve classification accuracy. From our results it is clear that all the three classifiers perform well with real, simulated and combined data. All these results show that the simulator we developed can be used to improve the performance of the real system. It can also be used as an alternative for the real system and simulate sensor outputs.

6.4 Summary

In this chapter we discussed the performance of various classifiers on the simulator dataset and the real dataset. For testing their performances we selected decision tree, multilayer perceptron and support vector machine classifiers. We used 10-fold cross validation and averaging techniques for training and testing purposes.

First we tested the real dataset for classification accuracy. We used 124 different datasets or profiles for our experiments. Also the tests were performed using only the 10-fold cross validation technique. The results obtained showed convincing performance of these algorithms in this domain.

We generated similar datasets using the simulator and ran the same set of experiments on the data generated by the simulator. The simulator dataset was also tested using 10-fold cross validation. It was found that the 10-fold cross validation technique performed well in the simulator domain also.

Then we combined the two datasets to generate a combined set of data.Then we ran the same set of experiments on this dataset too. We found out that the total number misclassifications were lesser with the combined data than the sum of misclassifications for real and simulator data. This proved that the classifiers perform well when they are trained with both real and simulator data.

In order to prove the statistical similarity between the two datasets we used t-test. We ran t-test on the classification accuracies obtained on each fold of the 10-fold cross validation tests. The t-test was performed between the accuracies of the real and simulator data. We obtained an average P value of greater than 0.05 on all our t-tests. And from this we inferred that our null-hypothesis can be true.

From all the experiments we performed, we concluded that the simulator can produce results similar to that of the real system. More research on the performance of the simulator would help us achieve better results.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

In this research we have studied the various security systems currently in existence. We investigated the latest trends in concealed weapon detection systems and also about advanced imaging systems. We first reviewed the sensor technologies being simulated in our research work. We discussed both passive imaging systems and active imaging systems and their potential for detecting and observing metallic and other ceramic objects concealed underneath common clothing. Recent advances in Millimeter wave sensor technology and infrared sensor technology have led to the development of many security systems for threat monitoring. However millimeter wave cameras or infrared cameras or metal detectors alone may not provide enough information about the threat being monitored or detected. In order to enhance the usefulness and threat detecting capability of the security systems, sensor fusion has been proposed as the best solution. By combining complementary information from different systems we can achieve better threat detection capability and security. Preliminary works on sensor fusion and threat detection have shown promising results and several systems have been developed recently. In our research work we decided to combine the outputs of millimeter wave sensor, infrared sensor, and metal detector sensor systems.

But building such a system would require considerable cost for deployment. Also before building the complete system we need to test the performance of the system under various different sets of parameters in a realistic environment at all phases of the development cycle. Simulation is a cost-effective choice for prototyping and testing these sensor systems. Simulation of these sensor systems can save cost, time, and complexity involved in deploying and constantly changing the input conditions for experimental purposes. Our goal is to develop a simulator that can simulate the outputs of a real security system with multiple sensors. In our research we simulated three different sensor systems: millimeter wave, infrared, and metal detector.

We first obtained the outputs of the real system for comparison purposes. Then the datasets were classified as threat and non-threat. The classification accuracy of the data is then tested using the classifiers in WEKA. We used support vector machine classifier, neural network classifier and decison tree classifier to test the datasets. Similar classification tests were then performed on the simulator dataset to get the classification accuracies on this dataset.

We also combined the two datasets to generate a combined set of data. Then we ran the same set of experiments on this dataset too. The results obtained using the real dataset and the simulated dataset were compared with the results of combined dataset to establish the accuracy of the simulator output. We found out that the total number misclassifications were lesser with the combined data than the sum of misclassifications for real and simulator data. This proved that the classifiers perform well when they are trained with both real and simulator data.

In order to prove the statistical similarity between the two datasets we also performed t-test. Using the t-test we tried to prove our null-hypothesis. We ran t-test on the classification accuracies obtained on each fold of the 10-fold cross validation tests. The t-test was performed between the accuracies of the real and simulator data. We obtained an average P value of greater than 0.05 on all our t-tests from which we inferred that there the differences between the two data sets could be due to chance only. In other words the results did not show a significant difference between the two distributions. We can also extend our results and show that the two distributions lie within some positive epsilon value of each other.

From all the experiments we performed, we concluded that the simulator can produce results similar to that of the real system. More research on the performance of the simulator would help us achieve better results.

From all the experiments we performed, we concluded that the simulator can produce results similar to that of the real system. More research on the performance of the simulator would help us achieve better results. The simulator can also be used as an alternative for testing the real system. Therefore we can perform more tests and generate more complex situations for testing purposes. This would help further developments in advanced security systems and homeland security.

7.2 FUTURE WORK

There are several challenges ahead in our field of research. Our simulator can be modified to simulate the responses of various other sensor systems. It can also be developed to simulate video sequences for any given situation. Also we could develop better image processing algorithms for better object recognition from the images. These simulated systems can then help to develop an advanced security system which combines various imaging techniques, combinations of sensor technologies and image processing techniques. The system should be capable of operating at a distance from the subject, with high probability of threat detection and low probability of false alarm. The system should also be capable of operating on its own and should be capable of detection a wide range of potential threats. Such a system can play a key role in addressing the today's security issues.

APPENDIX A

MILLIMETER WAVE SENSOR SYSTEM

A.1 Millivision Millimeter Wave Camera

For our research work we are using the passive millimeter wave camera Vela 125 concealed weapon detector from Millivision technologies. This millimeter wave detector uses passive millimeter waves to detect objects and concealed weapons. The system is capable of detecting drugs, plastics and explosives. It provides the users with the millimeter wave image of the person being viewed along with the visual image. This will help the users for better understanding of the situation. Regardless of a object being a threat or a non-threat this system provides the user with exact size, shape and the position information of the objects. Further processing of the output is required to determine whether the objects are a threat or not.

The Vela 125 camera is mounted on a tripod with a gear-driven adjustable base. The tripod will help the users to adjust the camera to their desired angle and orientation. The camera can be connected to store the output images directly to the system's local drive. The setting of the camera can also be adjusted using the software provided along with it. The default size of the output image from the camera is 28 x 28. The user can also control the noise, contrast and the sharpness of the output images. The table A.1 shows the specifications of the camera.

Table A.1. Specifications of Vela125 Millimeter Wave Camera

Illumination	Passive
Operating Frequency	91.5 GHz to 96.5 GHz
Lens Diameter	125 mm
Image Frame Rate	10 Hz
Raw Image Size	28 x 28 Pixels
Spatial Resoulution	1.5 cm at 1 Meter Range
Filed of View	0.43 x 0.43 Radians
Thermal Sensitivity	3 Kelvin r.m.s. minimum
Output	VGA and Raw Image

A.2 Images of Objects

Some images of noise patterns are shown in the following figures

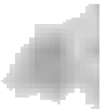


Figure A.1. Noise Pattern 1.

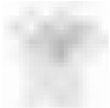


Figure A.2. Noise Pattern 2.

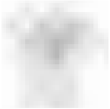


Figure A.3. Noise Pattern 3.

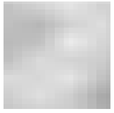


Figure A.4. Noise Pattern 4.

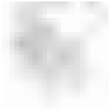


Figure A.5. Noise Pattern 5.

The following objects can be included in the output of our millimeter wave simulator.

(a) !ht (b) !ht

Figure A.6. (a) Horizontal Image of Gun (b) Vertical Image of Gun.

Figure A.7. Image of Iron Handle.

(a) !ht (b) !ht

Figure A.8. (a) Horizontal Image of Rod (b) Vertical Image of Rod.

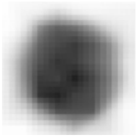


Figure A.9. Image of Compact Disk.

Figure A.10. Image of Metal Strip.

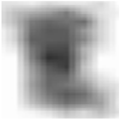


Figure A.11. Image of Metal Plate.

Figure A.12. Image of Metal Pendant.

Figure A.13. Image of Metal Barrel.

Figure A.14. Image of Belt Buckle.

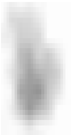


Figure A.15. Image of Cement Tile.

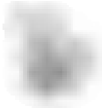


Figure A.16. Image of Ceramic Plate.

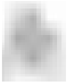


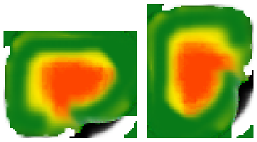
Figure A.17. Image of Mobile Phone.

APPENDIX B

INFRARED SENSOR SYSTEM

B.1 Images of Objects

The following objects can be included in the output of our infrared sensor simulator.



(a) h (b) h

Figure B.1. (a) Horizontal Image of Gun (b) Vertical Image of Gun.

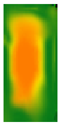


Figure B.2. Image of Iron Handle.

(a) h (b) h

Figure B.3. (a) Horizontal Image of Rod (b) Vertical Image of Rod.

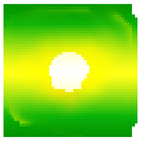


Figure B.4. Image of Compact Disk.

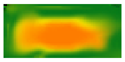


Figure B.5. Image of Metal Strip.

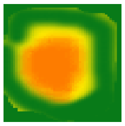


Figure B.6. Image of Metal Plate.

Figure B.7. Image of Metal Pendant.

Figure B.8. Image of Metal Barrel.

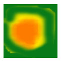


Figure B.9. Image of Belt Buckle.

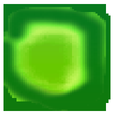


Figure B.10. Image of Ceramic Plate.

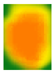


Figure B.11. Image of Mobile Phone.

APPENDIX C

METAL DETECTOR SYSTEM

C.1 Garret PD 6500i Metal Detector

For our research project we are using the PD 6500i metal detector from Garret metal detectors. The PD 6500i is a walk through metal detector. It uses the multi target pinpointing technology to detect and indicate the position of various kinds of metals present in the target. The detector has 33 zones that cover a target person from head to toe. The detector has a array of indicator lamps that show the region at which any metal is present. The system also has a alarm to indicate the presence of metals in the target. The PD 6500i also has a access control security panel to adjust the sensitivity settings of the detector. The control panel is password protected and only authorized personnel can change the settings of the detector.

The output of the PD 6500i detector is a set of 19 values. The default value of each of the 19 values is 200. Each of the 19 values corresponds to a certain region in the target's body and its value depends on the size and orientation of the metals in the target. The values may vary from 0-200 depending on the size of the metals. Lesser the value greater the amount metal present in the target. The table C.1 shows the specifications of the PD 6500i detector.

Table C.1. Specifications of PD 6500i Garret Metal Detector

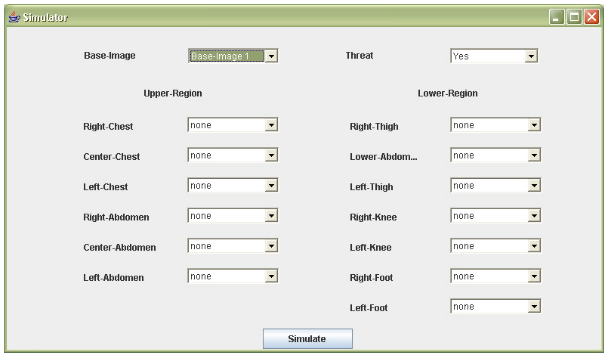
Operating Temperatures	-4F (-20C) to + 158F (70C)
Humidity	95% non-condensing
Output	19 Numeric Values ranging from 0-200

APPENDIX D

USER GUIDE

D.1 Graphical user Interface

Our simulator has a graphical user interface to make it easier for the user to select the input parameters during the simulation process. The GUI was designed used Java. The GUI is platform independent and can be run in any operating system. It helps the users to select an available object for the 13 regions in the body of a person. Figure D.1 shows a snapshot of the GUI.



Base-Image	Base-Image 1	Threat	Yes
Upper-Region		Lower-Region	
Right-Chest	none	Right-Thigh	none
Center-Chest	none	Lower-Abdom...	none
Left-Chest	none	Left-Thigh	none
Right-Abdomen	none	Right-Knee	none
Center-Abdomen	none	Left-Knee	none
Left-Abdomen	none	Right-Foot	none
		Left-Foot	none

Figure D.1. Graphical User Interface of The Simulator.

Before simulating an output the user can also select the category to which the current set of inputs belong i.e., threat or non-threat.

D.2 Steps For Simulation

The simulator is designed for a very simple operation. In order to simulate the millimeter wave, infrared and metal detector sensor outputs the user has to follow the instructions given below.

1] Copy the simulator files and save them in a folder in the local drive.

2] Open the command prompt and navigate to the folder containing the simulator files.

3] Now run the java file ”GUI.java” containing the GUI for the simulator.

4] Select one of base images for simulating the output images. There are five sets of different base images available with the simulator.

5] Select whether the given set of inputs is a threat or non-threat.

6] Select the objects for each region [13 regions: 6 upper and 7 lower] of the base image and press the simulate button to simulate the sensor outputs.

7] The program will create a text file ”threat.txt” which says whether the situation is an threat or non-threat, a text file ”metal.txt” containing the simulated metal detector output, ”upper.jpg” the simulated millimeter wave image of upper region of the body, ”lower.jpg” the simulated millimeter wave image of lower region of the body, and ”ir.jpg” the simulated infrared sensor output image in the output folder.

D.3 Steps For Generating the Feature Vectors

1] Open the command prompt and run the conversion program ”generate.java”.

2] Specify the folder containing all the simulated datasets as a command line parameter. The user can also specify the number of pixels to be averaged in the command line. The program uses the default pixel averaging values if it is not specified in the command line.

3] The result of the program is the feature vector representing both the millimeter wave sensor image and the metal detector sensor output.

D.4 Sample ARFF File

This is an ARFF input file for a set of simulated millimeter wave sensor images. The instances that belong to the positive class represent the threat and negatives represent the non-threat.

1. Title: Millimeter wave Sensor and Metal Detector Sensor Data

2. Source Information

Creator: Janakiram Natarajan, Sentry Group, University of Texas at Arlington

Date: February, 2006

3. Past Usage: None yet published

4. Relevant Information:

The instances were generated using the simulated Millimeter wave sensor and metal detector sensor outputs.

The images were segmented to create a classification for every pixel.

5. Number of Instances: Training data: 125 Test data: 125

6. Number of Attributes: 44 continuous attributes

7. Attribute Information: Attributes represent the sensor outputs

8. Missing Attribute Values: None

9. Class Distribution:

Classes: Positive, Negative.

Relabeled values in attribute class

From: 1 To: Positive

From: 2 To: Negative

@relation threat

@attribute region1 numeric

@attribute region2 numeric

@attribute region3 numeric

@attribute region4 numeric

@attribute region5 numeric

@attribute region6 numeric

@attribute region7 numeric

```
@attribute region8 numeric
@attribute region9 numeric
@attribute region10 numeric
@attribute region11 numeric
@attribute region12 numeric
@attribute region13 numeric
@attribute region14 numeric
@attribute region15 numeric
@attribute region16 numeric
@attribute region17 numeric
@attribute region18 numeric
@attribute region19 numeric
@attribute region20 numeric
@attribute region21 numeric
@attribute region22 numeric
@attribute region23 numeric
@attribute region24 numeric
@attribute region25 numeric
@attribute region26 numeric
@attribute region27 numeric
@attribute region28 numeric
@attribute region29 numeric
@attribute region30 numeric
@attribute region31 numeric
@attribute region32 numeric
@attribute region33 numeric
```

@attribute region34 numeric

@attribute region35 numeric

@attribute region36 numeric

@attribute region37 numeric

@attribute region38 numeric

@attribute region39 numeric

@attribute region40 numeric

@attribute region41 numeric

@attribute region42 numeric

@attribute region43 numeric

@attribute region44 numeric

@attribute class {*positive* , *negative* }

@data

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 212, 252, 239, 61, 0, 142, 249, 183, 33, 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200 , 200, negative

76, 53, 180, 51, 55, 0, 71, 230, 134, 15, 0, 183, 237, 234, 39, 0, 211, 250, 239, 61, 0, 142, 249, 183, 33, 158, 1, 2, 2, 22, 47, 78, 200, 200, 199, 200, 7, 7, 30, 11, 5, 15, 200, 200, positive

6, 53, 180, 51, 55, 0, 71, 228, 134, 15, 0, 187, 234, 234, 39, 0, 212, 250, 239, 61, 0, 142, 249, 183, 33, 158, 1, 2, 19, 13, 39, 73, 200, 200, 199, 200, 4, 4, 30, 6, 3, 18, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 232, 131, 15, 0, 187, 250, 217, 39, 0, 212, 252, 236, 61, 0, 142, 249, 183, 33, 158, 1, 1, 2, 7, 35, 69, 200, 200, 199, 200, 2, 2, 30, 2, 1, 22, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 209, 244, 239, 61, 0, 142, 243, 183, 33, 193, 112, 75, 2, 1, 1, 96, 200, 200, 190, 196, 53, 53, 94, 64, 82, 53, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 212, 239, 239, 61, 0, 142, 244, 183, 33, 193, 112, 75, 5, 4, 4, 94, 200, 200, 190, 195, 43, 43, 84, 62, 77, 52, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 212, 252, 228, 61, 0, 142, 248, 178, 33, 193, 112, 75, 7, 6, 6, 92, 200, 200, 190, 195, 43, 43, 84, 62, 77, 52, 200, 200, positive

76, 53, 180, 51, 55, 0, 70, 231, 134, 15, 0, 183, 237, 234, 39, 0, 210, 240, 239, 61, 0, 141, 242, 183, 33, 158, 1, 1, 1, 1, 1, 43, 200, 200, 196, 200, 1, 1, 1, 1, 1, 1, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 232, 134, 15, 0, 187, 224, 233, 39, 0, 212, 251, 233, 61, 0, 142, 248, 182, 33, 158, 1, 1, 1, 1, 1, 42, 200, 200, 199, 198, 1, 1, 1, 1, 1, 8, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 232, 131, 15, 0, 183, 241, 217, 39, 0, 212, 252, 236, 61, 0, 142, 249, 183, 33, 158, 1, 1, 1, 1, 1, 43, 200, 200, 198, 200, 1, 1, 22, 1, 1, 5, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 232, 132, 15, 0, 187, 245, 202, 39, 0, 209, 244, 236, 61, 0, 142, 243, 183, 33, 193, 112, 75, 2, 1, 1, 96, 200, 200, 190, 196, 53, 53, 94, 64, 82, 53, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 182, 231, 234, 39, 0, 212, 235, 238, 61, 0, 142, 246, 183, 33, 192, 70, 17, 1, 1, 1, 63, 200, 200, 190, 195, 1, 1, 23, 9, 22, 32, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 232, 134, 15, 0, 187, 218, 234, 39, 0, 212, 252, 228, 61, 0, 142, 248, 178, 33, 192, 64, 15, 1, 1, 1, 61, 200, 200, 190, 195, 1, 1, 25, 1, 7, 34, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 176, 224, 234, 39, 0, 212, 250, 239, 61, 0, 142, 249, 183, 33, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 239, 234, 39, 0, 212, 232, 239, 61, 0, 142, 241, 183, 33, 200, 153, 138, 141, 163, 165, 174, 200, 199, 199, 200, 131, 131, 192, 159, 165, 186, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 225, 39, 0, 212, 252, 237, 61, 0, 142, 249, 183, 33, 199, 148, 138, 132, 160, 173, 170, 200, 200, 200, 200, 103, 103, 143, 118, 118, 187, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 208, 240, 233, 61, 0, 140, 240, 182, 33, 200, 192, 185, 140, 136, 147, 169, 200, 200, 200, 198, 169, 169, 160, 141, 181, 190, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 187, 242, 234, 39, 0, 212, 248, 239, 61, 0, 142, 246, 183, 33, 178, 78, 66, 22, 30, 53, 106, 200, 200, 198, 199, 8, 8, 24, 12, 1, 54, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 212, 250, 225, 61, 0, 142, 247, 174, 33, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 232, 134, 15, 0, 186, 242, 234, 39, 0, 212, 252, 239, 61, 0, 142, 249, 183, 33, 178, 85, 72, 70, 97, 115, 139, 200, 200, 199, 200, 71, 71, 85, 79, 75, 97, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 187, 242, 234, 39, 0, 212, 249, 239, 61, 0, 142, 245, 183, 33, 178, 79, 66, 26, 30, 52, 104, 200, 200, 196, 200, 15, 15, 30, 13, 1, 54, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 207, 243, 239, 61, 0, 141, 245, 183, 33, 200, 199, 197, 159, 146, 142, 165, 200, 200, 197, 200, 165, 165, 145, 150, 138, 155, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 252, 234, 39, 0, 212, 243, 225, 61, 0, 142, 247, 174, 33, 200, 192, 185, 139, 134, 144, 172, 200, 200, 200, 198, 174, 174, 165, 171, 183, 191, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 176, 224, 234, 39, 0, 212, 250, 235, 61, 0, 142, 249, 180, 33, 200, 198, 196, 151, 146, 144, 168, 200, 200, 200, 199, 150, 150, 133, 148, 131, 156, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 187, 242, 234, 39, 0, 212, 234, 239, 61, 0, 142, 247, 183, 33, 178, 78, 66, 22, 30, 53, 106, 200, 200, 198, 199, 8, 8, 24, 12, 1, 54, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 232, 134, 15, 0, 186, 233, 213, 39, 0, 212, 251, 239, 61, 0, 142, 249, 183, 33, 136, 1, 1, 1, 1, 1, 8, 200, 200, 198, 200, 1, 1, 1, 1, 1, 1, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 183, 243, 234, 39, 0, 212, 238, 238, 61, 0, 142, 246, 183, 33, 193, 70, 15, 1, 1, 1, 68, 200, 200, 189, 195, 1, 1, 76, 36, 57, 35, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 231, 134, 15, 0, 187, 239, 223, 39, 0, 212, 238, 239, 61, 0, 142, 243, 183, 33, 178, 27, 2, 1, 1, 18, 80, 200, 199, 198, 199, 1, 1, 16, 1, 1, 44, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 228, 134, 15, 0, 187, 234, 234, 39, 0, 212, 247, 225, 61, 0, 142, 247, 174, 33, 158, 1, 2, 19, 13, 39, 73, 200, 200, 199, 200, 4, 4, 30, 6, 3, 18, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 230, 134, 15, 0, 183, 237, 234, 39, 0, 211, 247, 239, 61, 0, 142, 245, 183, 33, 158, 1, 1, 1, 1, 1, 43, 200, 200, 196, 200, 1, 1, 1, 1, 1, 1, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 229, 134, 15, 0, 187, 213, 234, 39, 0, 212, 248, 229, 61, 0, 142, 246, 181, 33, 193, 112, 75, 7, 6, 6, 92, 200, 200, 190, 195, 43, 43, 84, 62, 77, 52, 200, 200, positive

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 249, 223, 39, 0, 212, 252, 239, 61, 0, 142, 249, 183, 33, 200, 149, 136, 139, 158, 165, 174, 200, 199, 200, 200, 120, 120, 192, 144, 151, 190, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 232, 134, 15, 0, 187, 243, 234, 39, 0, 208, 239, 239, 61, 0, 140, 240, 183, 33, 199, 152, 140, 135, 165, 177, 169, 200, 200, 200, 200, 119, 119, 141, 131, 130, 182, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 232, 131, 15, 0, 183, 241, 217, 39, 0, 212, 249, 225, 61, 0, 142, 247, 179, 33, 158, 1, 1, 1, 1, 1, 11, 200, 200, 198, 199, 1, 1, 1, 1, 1, 1, 200, 200, positive

76, 53, 180, 51, 56, 0, 71, 232, 134, 15, 0, 187, 218, 234, 39, 0, 212, 232, 239, 61, 0, 142, 241, 183, 33, 199, 152, 140, 135, 165, 177, 169, 200, 200, 200, 200, 119, 119, 141, 131, 130, 182, 200, 200, negative

76, 53, 180, 51, 55, 0, 71, 233, 134, 15, 0, 187, 239, 234, 39, 0, 208, 241, 235, 61, 0, 140, 240, 180, 33, 200, 151, 134, 92, 109, 109, 142, 200, 199, 199, 199, 81, 81, 125, 107, 96, 142, 200, 200, negative

REFERENCES

[1] Brijot imaging systems. *http://www.brijot.com/sysoverview.php.*

[2] Idaho national laboratory. *http://citeseer.ist.psu.edu/599648.html.*

[3] Transportation security administration. *http://www.tsa.gov/public/display?theme=70.*

[4] S. Alhakeem and P. K. Varshney. Decentralized bayesian hypothesis testing with feedback. *IEEE Transactions Systems, Man Cybern.*, pages 503–513, July 1996.

[5] P. H. Bernd Heisele and T. Poggio. Face recongnition with support vector machines: Global versus component-based approach. 2001.

[6] K. S. Blum, R.S. and H. Poor. Distributed detection with multiple sensors ii. advanced topics. In *Proceedings of the IEEE*, number 5494511, pages 64 – 79, Jan. 1997.

[7] H. M. Chen and P. K. Varshney. Automatic two-stage IR and MMW image registration algorithm for concealed weapons detection. *IEE Proceedings-Vision Image and Signal Processing*, 148(4):209–216, Aug. 2001.

[8] H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. Technical Report MS-CIS-86-08, University of Pennsylvania.

[9] M. Fardanesh and O. Ersoy. Classification accuracy improvement of neural network classifiers by using unlabeled data. In *Geoscience and Remote Sensing, IEEE Transactions*, number 0196-2892, pages 1020 – 1025, May 1998.

[10] G. D. Guo, S. Z. Li, and K. L. Chan. Face recognition by support vector machines. In *International Conference on Automatic Face and Gesture Recognition*, pages 196–201, 2000.

[11] R. M. R. M.-A. S. Hua-Mei Chen, Seungsin Lee and P. K. Varshney. Imaging for concealed weapon detection. *IEEE Signal processing magazine*, (1053-5888), Mar. 2005.

[12] D. G. R. Huguenin. The detection of hazards and screening for concealed weapons with passive millimeter wave imaging concealed threat detectors. 2004.

[13] R. D. King, C. Feng, and A. Sutherland. STALOG: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):289–333, 1995.

[14] S. Laurence, I. Burns, A. Back, and A. C. Tsoi. Neural network classification and prior class probabilities. *j-LECT-NOTES-COMP-SCI*, 1524:299–, 1998.

[15] C. C. Lee and J. J. Chao. Optimum local decision space partioning for distributed detection. *IEEE Trans. Aerospace Elect. Syst*, pages 536–544, 1989.

[16] S. Lee, R. Rao, and M.-A. Slamani. Noise reduction and object enhancement in passive millimeter wave concealed weapon detection. In *ICIP (1)*, pages 509–512, 2002.

[17] Z. Liu, Z. Xue, R. Blum, and R. Laganière. Concealed weapon detection and visualization in a synthesized image. *Pattern Anal. Appl*, 8(4):375–389, 2006.

[18] R. Maree, P. Geurts, G. Visimberga, J. Piater, and L. Wehenkel. A comparison of generic machine learning algorithms for image classification. *IEEE Signal processing magazine*, 2004.

[19] C. N. F. D. J. McMillan, R.W. and M. Wicks. Concealed weapon detection using microwave and millimeter wave sensors. In *Microwave and Millimeter Wave Technology Proceedings, 1998. ICMMT '98.*, number 6239422, pages 1 – 4, Aug. 1998.

[20] O. Minin. Quasioptical mm-wave concealed weapon detection systems: brief review. In *Antennas, Propagation and EM Theory, 2003. Proceedings. 2003 6th International SYmposium*, number 8046293, pages 356 – 358, Nov. 2003.

[21] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR*, pages 130–136, 1997.

[22] K. Pados, D.and Halford, D. Kazakos, and P. Papantoni-Kazakos. Distributed binary hypothesis testing with feedback. In *Systems, Man and Cybernetics, IEEE Transactions*, pages 21 – 42, Jan. 1995.

[23] G. A. Ramirez and O. Fuentes. Face detection using combinations of classifiers. In *CRV*, pages 610–615, 2005.

[24] M. D. Sheen, D.M. and T. Hall. Three-dimensional millimeter-wave imaging for concealed weapon detection. In *Microwave Theory and Techniques, IEEE Transactions*, number 0018-9480, pages 1581 – 1592, Sept. 2001.

[25] X. Shi. A study on bayes feature fusion for image classification, Aug. 18 2003.

[26] S. D. Silverstein and Y. Zheng. 3-D image reconstruction from near-field coherently scattered waves. In *ICIP (3)*, pages 161–164, 2003.

[27] M.-A. Slamani, P. K. Varshney, R. M. Rao, M. G. Alford, and D. Ferris. Image processing tools for the enhancement of concealed weapon detection. In *ICIP (3)*, pages 518–522, 1999.

[28] C. L. Smith. The security systems research and testing laboratory at edith cowan university. In *Security Technology, 2005. CCST '05. 39th Annual 2005 International Carnahan Conference*, pages 4 – 7, Oct. 2005.

[29] S. Sundresh, W. Kim, and G. Agha. SENS: A sensor, environment and network simulator. In *Annual Simulation Symposium*, page 221, 2004.

[30] P. Svensson and P. Horling. Building an information fusion demonstrator. In *Information Fusion, 2003. Proceedings of the Sixth International Conference*, pages 1316 – 1323, July 2003.

[31] P. Swaszek and P. Willett. Parley as an approach to distributed detection. In *Aerospace and Electronic Systems, IEEE Transactions*, pages 447 – 457, Jan. 1995.

[32] S. Thomopoulos, R. Viswanathan, and D. Bougoulias. Optimal distributed decision fusion. In *Aerospace and Electronic Systems, IEEE Transactions*, pages 761 – 765, 1989.

[33] J. N. Tsistsiklis, H. V. Poor, and J. B. Thomas. Decentralized detection. *Advances in Statistical Signal Processing, Signal Detection, vol. 2,*, 1993.

[34] P. K. Varshney, H.-M. Chen, L. C. Ramac, and M. Uner. Registration and fusion of infrared and millimeter wave images for concealed weapon detection. In *Proceedings of the 1999 International Conference on Image Processing (ICIP-99)*, pages 532–536, Los Alamitos, CA, Oct. 24–28 1999. IEEE.

[35] P. K. Varshney, H. M. Chen, L. C. Ramac, M. Uner, D. Ferris, and M. Alford. Registration and fusion of infrared and millimeter wave images for concealed weapon detection. In *International Conference on Image Processing*, pages III:532–536, 1999.

[36] R. Viswanathan and P. Varshney. Distributed detection with multiple sensors i. fundamentals. In *Proceedings of the IEEE*, number 5494510, pages 54 – 63, Jan. 1997.

[37] P. L. Vora, J. E. Farrell, J. D. Tietz, and D. H. Brainard. Image capture: simulation of sensor responses from hyperspectral images. *IEEE Transactions on Image Processing*, 10(2):307–316, 2001.

[38] D. Wilson. Use of modeling and simulation to support airport security. *IEEE Aerospace and Electronic Systems Magazine*, 20 , Issue: 8 , Part 1(0885-8985):3 – 6, Aug. 2005.

[39] L. Wuyan and Yangwanhai. Optimal distributed decision fusion in the sense of the neyman-pearson test. In *Radar, 2001 CIE International Conference on, Proceedings*, number 7321121, pages 708 – 712, Oct. 2001.

[40] Z. Xue and R. Blum. Concealed weapon detection using color image. In *Information Fusion, 2003. Proceedings of the Sixth International Conference*, pages 622 – 627, Oct. 2003.

[41] B. Xue.Z and Li.Y. Fusion of visual and ir images for concealed weapon detection. In *Information Fusion, 2002. Proceedings of the Fifth International Conference*, number 7412204, pages 1198 – 1205 vol.2, July 2002.

[42] J. Yang and R. S. Blum. A statistical signal processing approach to image fusion for concealed weapon detection. In *ICIP (1)*, pages 513–516, 2002.

[43] K. R. P. Z. B. Tang and D. L. Kleinman. Optimization of distributed detection networks: Part ii generalized tree structures. In *IEEE Trans. Syst., Man Cybern., vol 23*, pages 211–221, Feb. 1993.

[44] Z. Zhang and R. S. Blum. Region-based image fusion scheme for concealed weapon detection, Mar.06 2006.

BIOGRAPHICAL INFORMATION

Janakiram Natarajan was born in Dindigul, India in May 1982. He received his Bachelors of Engineering in Computer Science and Engineering from the University of Madras in March 2003.

He began his study toward the Masters degree in the department of Computer Science and Engineering at the University of Texas at Arlington in August 2003. He worked as a Research Assistant for Dr. Lawrence Holder, University of Texas at Arlington.

He received his Masters in Computer Science and Engineering from the University of Texas at Arlington, Texas, in May 2006. His research interests focus on artificial intelligence, machine learning, security and image processing.