A LIGHTWEIGHT WIRELESS MULTIMEDIA SENSOR NETWORK ARCHITECTURE WITH OBJECT DETECTION AND CLASSIFICATION CAPABILITY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUHSİN CİVELEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

FEBRUARY 2017

Approval of the thesis:

A LIGHTWEIGHT WIRELESS MULTIMEDIA SENSOR NETWORK ARCHITECTURE WITH OBJECT DETECTION AND CLASSIFICATION CAPABILITY

submitted by **MUHSİN CİVELEK** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ____________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** ____________________

Prof. Dr. Adnan Yazıcı
Supervisor, **Computer Engineering Department, METU** ____________________

Examining Committee Members:

Prof. Dr. Ahmet Coşar
Computer Engineering, METU ____________________

Prof. Dr. Adnan Yazıcı
Computer Engineering, METU ____________________

Assoc.Prof. Dr. Sinan Kalkan
Computer Engineering, METU ____________________

Assoc.Prof. Dr. Murat Koyuncu
Information Systems Engineering, Atılım University ____________________

Asst.Prof.Dr. Mustafa Sert
Computer Engineering, Başkent University ____________________

Date: 03.02.2017

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MUHSİN CİVELEK

Signature :

ABSTRACT

A LIGHTWEIGHT WIRELESS MULTIMEDIA SENSOR NETWORK ARCHITECTURE WITH OBJECT DETECTION AND CLASSIFICATION CAPABILITIES

Civelek, Muhsin
Ph.D., Department of Computer Engineering
Supervisor : Prof. Dr. Adnan Yazıcı

February 2017, 99 Pages

Use of wireless multimedia sensor networks (WMSNs) for surveillance applications has attracted the interest of many researchers. As with traditional sensor networks, it is easy to deploy and operate WMSNs. With inclusion of multimedia devices in wireless sensor networks (WSNs), it is possible to provide data to users that is more meaningful than that provided by scalar sensor-based systems alone; however, producing, storing, processing, analyzing, and transmitting multimedia data in sensor networks requires consideration of additional constraints, including energy, processing power, storage capacity, and communication. Furthermore, as multimedia sensors produce much more data than scalar sensors, more manpower is required to analyze multimedia data. To overcome these constraints and challenges, this study aimed to propose a system architecture and a set of procedures for WMSNs that facilitate automatic classification of moving objects using scalar and multimedia sensors. Methods and standards for detecting and classifying a moving object, as well as transmission of the results, are described in detail. The hardware for each sensor node includes a built-in camera, a passive infrared motion sensor, a vibration sensor, and an acoustic sensor. An application using the proposed methods was developed and embedded in the multimedia sensor node. In addition, a sink station was setup and the data produced by the sensor network was collected by this server. The classification performance of the application was tested using video recorded by the sensor node. The effect of the proposed methods on power consumption was also tested and measured. The experimental results show that the proposed approach is sufficiently lightweight to be used for real-world surveillance applications.

Keywords: Wireless Multimedia Sensor Network, Object detection, Object Classification

ÖZ

NESNE TESPİT VE SINIFLANDIRMA YETENEĞİNE SAHİP HAFİF KABLOSUZ ÇOKLU ORTAM DUYARGA AĞI MİMARİSİ

Civelek, Muhsin
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Adnan Yazıcı

Şubat 2017, 99 Sayfa

Kablosuz çoklu ortam sensor ağlarının (KÇSA) gözetleme uygulamaları için kullanımı birçok araştırmacının ilgisini çekmektedir. Geleneksel sensor ağlarında olduğu gibi, KÇSA kurulumu ve işletilmesi kolay sistemlerdir. Çoklu ortam cihazlarının kablosuz sensor ağlarına dahil edilmesi, sadece skalar sensörlerden oluşan sistemlere nazaran daha anlamlı bilgilerin kullanıcılar için üretilmesine olanak sağlamıştır; ancak sensor ağlarında çoklu ortam verisinin üretimi, saklanması, işlenmesi, analiz edilmesi ve transferi enerji, işlem gücü, depolama kapasitesi ve iletişim ortamı ile ilgili ilave kısıtların dikkate alınmasını gerektirmektedir. Bunula beraber çoklu ortam sensörleri skalar sensörlere oranla çok daha fazla veri üreteceğinden bu verinin analizi daha fazla insan gücü gerektirecektir. Bahsedilen zorluk ve kısıtlarla başa çıkabilmek için bu çalışmada KÇSA hareket halindeki nesnelerin skalar ve çoklu ortam sensörleri yardımı ile otomatik sınıflandırmasını sağlayacak bir mimari ve buna işlevler kümesi önerilmektedir. Hareket eden nesnelerin sınıflandırılması ve sınıflandırma sonuçlarının iletilmesi için önerilen metot ve standartlar ayrıntılı olarak tanımlanmıştır. Çalışmamız kapsamında oluşturduğumuz sensor düğümlerinde kamera, pasif kızılötesi hareket sensörü, titreşim sensörü ve akustik sensor yer almaktadır. Önerdiğimiz metotları içeren bir uygulama da geliştirilerek çoklu ortam sensor düğümlerine gömülmüş durumdadır. Bunlara ilave olarak sensör ağından verileri toplama için bir sunucu sistemi kurulmuştur. Geliştirilen uygulamanın sınıflandırma başarısı sensor düğümü ile kaydedilen video dosyaları üzerinde test edilmiştir. Önerilen yöntemlerin güç tasarrufuna olan katkısı da ayrıca test edilerek ölçülmüştür. Deney sonuçları önerilen yaklaşımın gerçek gözetleme uygulamaları için yeterince hafif olduğunu ortaya koymaktadır.

Anahtar Kelimeler: Kablosuz Çoklu Ortam sensor Ağları, Nesne Tespiti, Nesne Sınıflandırma

ACKNOWLEDGMENTS

I would like to express my inmost gratitude to my supervisor Prof. Dr. Adnan Yazıcı for his guidance, encouragement, insight throughout the research and trust on me. It is an honor for me to share his knowledge and wisdom.

Additionally, I am grateful to Assoc. Prof. Dr. Murat Koyuncu, Assoc. Prof. Dr. Sinan Kalkan and Asst. Prof. Dr. Mustafa Sert for their valuable guidance, support and advices that steered me in the right the direction whenever I needed.

I would also like to thank to Prof.Dr.Ahmet COŞAR. The door to Prof. Coşar's office was always open whenever I ran into a trouble spot or had a question about my research or writing.

Additionally, I am grateful to Saeid Sattari for his support to annotate the videos which are used at the experiments.

I am indebted to my family for all their support and self-sacrifice on my behalf.

This dissertation was supported in part by a research grant from TÜBİTAK (The Scientific and Technological Research Council of Turkey) with the grant number 114R082.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Use of sensor technology for security and surveillance purposes is important for many traditional applications, ranging from civil and military applications to modern healthcare [1]. Monitoring a patient's physiological data, detection of foreign chemical agents in the air and water [2], and securing indoor and outdoor areas using video cameras and passive infrared (PIR) sensors are examples of such applications. The main problem associated with traditional surveillance systems that use simple scalar results and multimedia data is that they require a high degree of operator intervention to fuse and interpret the data. In addition, they have high false positivity rates and limited ability to produce meaningful results, and most of the time they are not scalable.

Traditional systems remain the preferred method for surveillance applications because they are easy and inexpensive to setup and operate, however, the need for sensor networks that require a minimum of human intervention to operate and that are capable of interpreting events is increasing. Development of intelligent, scalable, easily deployable, and long-life sensor networks is the focus of current research, especially for security applications. The necessity of such surveillance is increasing, particularly in instances when assigning humans to the task is nearly impossible or unfeasible due to environmental conditions. What is required for such conditions is lightweight wireless sensor surveillance networks that interpret the data they collect, and subsequently make conclusions and take action

A WMSN is composed of wireless scalar and multimedia sensor hardware to collect data from an environment as well as processing and communication units to fuse and examine the collected data and to transfer results. Since the network is composed of wireless elements, there exist additional constraints compared to wired systems. First of all, wireless sensors that are powered by batteries have to operate for an acceptable lifetime. All tasks that are performed within the network should consume energy efficiently. Second, the sensor nodes have limited processing capability. Therefore, the complexity of the tasks that are executed at the sensor nodes would also be limited. Communication is yet another important constraint that effects both power consumption and the quality of transferred data. The sensor nodes should be able to communicate using low bandwidth links and the amount of transferred data should be kept minimal to decrease power consumption of transmit and receive operations.

Sensor node hardware is evolving with the improvements in various technological fields like microcontroller, communication and sensing hardware etc. Utilization of multimedia devices in surveillance systems causes more reliable and meaningful data production. In addition, recent advances in the battery and wireless communication technologies allow the sensors to operate standalone for very long durations. Those improvements result in easily deployable and portable surveillance networks. Nonetheless, all of those improvements do not reduce dependency on human operators to interpret data collected by security systems. Making conclusions based on the collected data and deciding what actions are needed remain the responsibility of staff working as part of the surveillance system.

Following advancements in sensor technology, it is now possible to build intelligent surveillance systems with the addition of processor and memory technologies to the network. Wireless multimedia sensor networks consisting of heterogeneous sensor devices are now able to process and interpret their collected data using their hardware and software components. Smart security systems designed in this way can be used when staff assignment is difficult or dangerous.

Research on efficient and effective use of WMSN technology has focused on various sub domains. Some studies have investigated implementation of energy efficient multimedia sensor node hardware [3]-[9]. Mote hardware, communication modules, sensor hardware, and software components are integrated to build intelligent and energy-efficient sensor nodes with reasonable performance and cost efficiency. Earlier research primarily focused on sensor network architecture [10], [11]. Multi-tier sensor network architecture is generally proposed for and used within the scope of such research. Additionally, some studies have investigated such specific issues as data fusion, communication/routing, compression, encoding, and power consumption.

Some researchers have studied WMSN applications [12]-[15]. Surveillance and environmental monitoring systems that include low power scalar and video sensors are the primary use of WMSN technology. Typically, such surveillance systems consist of smart nodes that perform in-network processing, as well as sink stations that collect and fuse the data coming from the network.

Object detection and recognition are key capabilities of a surveillance system to consider it an intelligent system. Numerous studies on the application of object detection in wired video surveillance networks have been published [16]-[22]. These types of networks perform their analysis in real time using high-resolution video streams. Methods that require high processing power and massive data storage and memory space can be used in such applications. Implementation of object detection and recognition using WMSNs are more difficult than that using wired and stable surveillance networks because of the constraints stated above.

Our study aims to propose the design of a lightweight WMSN architecture for surveillance applications. The proposed architecture is capable of detection and recognition of threats without human operator intervention. It fulfills its function by taking advantage of both scalar and multimedia sensors. The lifetime of this lightweight network is also prolonged due to the avoidance of continuous multimedia streaming, thereby reducing the cost of communication, in terms of power consumption.

Starting from the network architecture, node hardware, communication within the network, object recognition capability, and interfaces between the sensor network and the system operators were implemented. The most common methods of automatic object identification in WMSNs were also investigated. The features and the methods used to classify objects were examined. Implementation of the proposed architecture was also demonstrated using sensor node hardware specifically designed for this study.

In addition, a number of experiments were performed to test the recognition performance of the application deployed on the sensor node hardware. These experiments were performed using videos recorded by the sensor node. Furthermore, the effect of the proposed system on power consumption was measured. It was observed that the proposed methods not only yielded reasonable object recognition accuracy, but were also energy efficient. The findings show that by using the proposed architecture and methods it is possible to implement an outdoor surveillance sensor network that is capable of detecting and identifying threats without human intervention. The proposed network is capable of producing simple text-based as well as multimedia output, and distributes that output in an energy-efficient manner, prolonging the life of the system. In contrast to earlier studies on WMSNs, we propose a complete WMSN system architecture that is capable of processing its own data, and making conclusions and distributing them.

1.1 Contributions

The WMSN architecture proposed in the present study provides 2 (two) primary contributions to WMSNs. The first is related to the lifetime of the network. One of the main constraints of wireless sensor networks is power consumption, most of which is consumed during communication between sensor nodes. The system proposed herein uses in-network processing. Rather than streaming large quantities of multimedia data, text-based object recognition results are used, which considerably decreases the quantity of transferred data, decreases the cost of communication, and prolongs the life of the sensor node. Furthermore, the proposed architecture still facilitates transmission of complex data, such as a silhouette or picture, whenever required.

The second contribution of the present study is related to operation of the system. The system proposed herein uses an intelligent lightweight WMSN capable of making its own decisions based on analysis and interpretation of scalar and multimedia data. The system does not require human operators to evaluate data and analyze threats. As the proposed system eliminates the need for human operators, it is possible to setup the system in areas where assigning staff is unfeasible or impossible.

1.2 Organization of the Thesis

The rest of this dissertation is organized as follows. In Chapter 2, the background of the related methods and technologies and review of the similar studies in the literature are introduced. Chapter 3 presents the proposed WMSN architecture. In Chapter 4 we give details of the architecture implementation. The experiments and evaluation of test results of the proposed system are presented in Chapter 5. Chapter 6 includes the conclusion and suggested additional research.

CHAPTER 2

BACKGROUND AND RELATED WORK

Detection and classification of a threat by WMSNs include several steps. Methods that are used to implement those steps vary according to network architecture, software, hardware components, environmental conditions, and constraints of the network. As such networks are composed of wireless sensor devices, power consumption, processing capability, communication, and storage constraints should be considered carefully during the selection and implementation of each method. Common WMNS architectures, its components and major techniques used for threat classification applications are discussed in this section.

2.1 WMSN Architecture

WMSNs are composed of audio/video sensing devices that are capable of retrieving and transmitting multimedia content such as audio, video and still images. Furthermore, such sensing devices can capture temperature, humidity, light intensity, and many other environment-related readings [23]. Some of the sensors that are operating within the network have also processing capability which enables implementation of further operations like data fusion, image segmentation and analysis, object classification etc.

There are 3 (three) main types of WMSN architecture as shown in Fig.2.1 [1]. The first type is the single-tier flat architecture, which consists of homogenous sensor nodes with similar functions and capabilities [24]. Sensor nodes send their data directly to the sink in this architecture using the selected network protocol. If each node has its own processing power, then distributed in-network processing can be performed, which prolongs the life of the network.

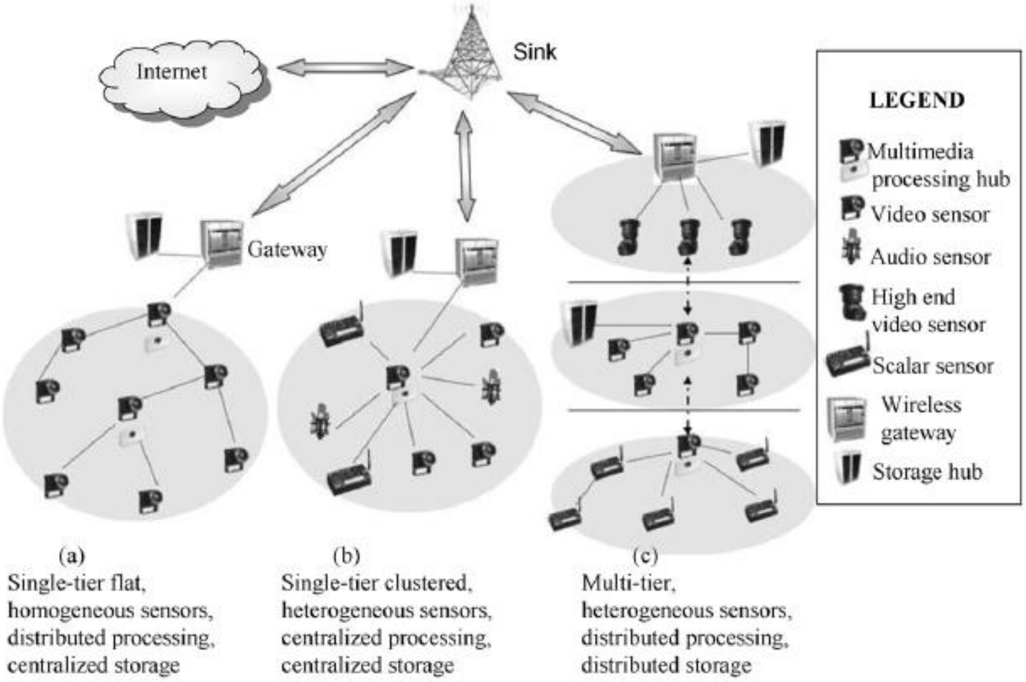

Internet
Sink
Gateway
LEGEND
Multimedia processing hub
Video sensor
Audio sensor
High end video sensor
Scalar sensor
Wireless gateway
Storage hub
(a)
Single-tier flat, homogeneous sensors, distributed processing, centralized storage
(b)
Single-tier clustered, heterogeneous sensors, centralized processing, centralized storage
(c)
Multi-tier, heterogeneous sensors, distributed processing, distributed storage

Figure 2.1 WMSN reference architecture (Figure Source [1])

Second approach is single tier clustered architecture which consists of heterogeneous sensor nodes with different capabilities. The sensor nodes in the cluster gather scalar as well as multimedia information and sends it to the cluster head which act as central processing unit for that cluster (having more resources and computational power as compared to other cluster nodes) [23]. The communication to the sink is either via the cluster head or a gateway which is connected to the cluster head wirelessly

The third approach is the multi-tier architecture, which consists of several layers of homogenous sensor nodes within this layer. In this architecture type the layers are heterogeneous. The first layer is composed of scalar sensors that are not able to generate multimedia data and do not have processing capability. This layer performs simple sensing tasks, such as sensing motion, audio, and pressure. Sensors in this layer are connected to a more powerful sensor in the upper layer. It is possible to add more layers that provide additional capabilities to the system. In addition, the upper-most layer is connected to the sink directly or via a gateway.

2.1.1 Sensor Structure

Sensors' functional components vary according to its capabilities and sensed data. Main block of common components are shown in Fig.2.2. The sensor part is mainly a transducer which collects data from the environment and forwards it to the converter if necessary. For this part, the availability of complementary metal-oxide semiconductor (CMOS) camera and small microphones make possible the development of WMSNs capable of gathering the multimedia information from the surrounding environment [23]. Micro-controller part contains processors with various scales and internal memory modules. According to the function of the node the adequate processing power is provided from this part. The transceiver unit is responsible for network communication. And power source provides necessary energy for the operation of all other parts. If sensor is planned to hold limited amount of information then an external storage (usually flash memory module) is integrated with it.

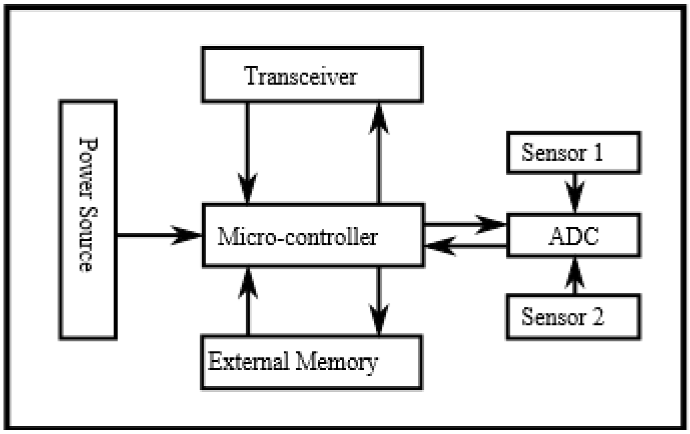


Figure 2.2 Typical sensor node (Figure Source [23])

It would be convenient to classify the sensor nodes according to its capabilities and module elements. Fig.2.3 represents such a classification [24]. Looking at the leaves of the tree, the lightweight motes can be considered as scalar sensors which collect data from environment with minimum power usage and minimum processing and storage capability. Their communication interface is designed to keep power consumption minimum. Intermediate class of motes has better processing and storage capacities than the previous ones. They may also contain camera modules in order to support video applications. The personal digital assistant (PDA) class motes have more processing power, better communication interfaces and larger storage areas. Those modules contain operating systems and mostly used to process multimedia data. However their power usage is significantly higher than the other classes. Those sensor nodes mostly form the first tier of multitier WMSN architecture. Sample motes are given with their properties in Table 2.1.

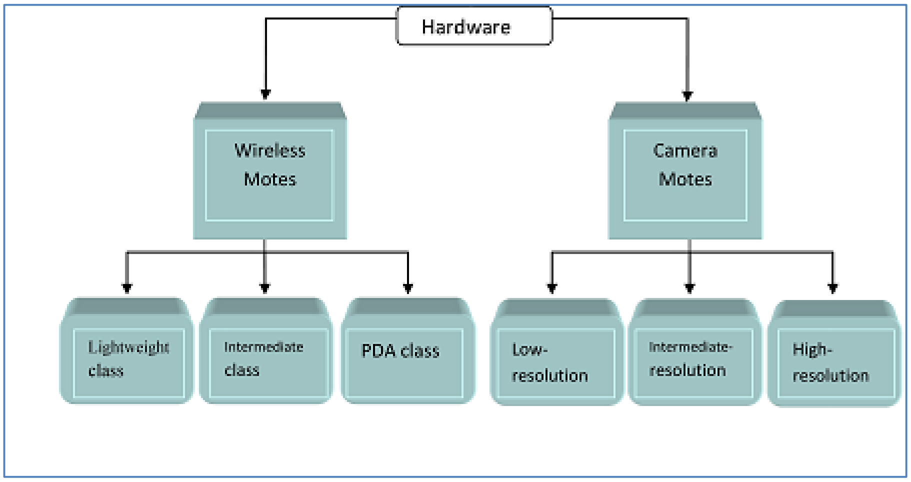
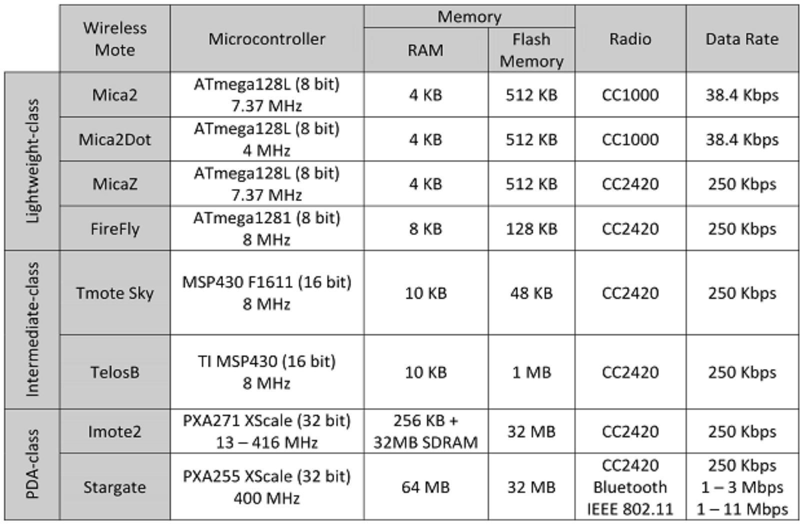


Figure 2.3 Mote classification (Figure Source [24])

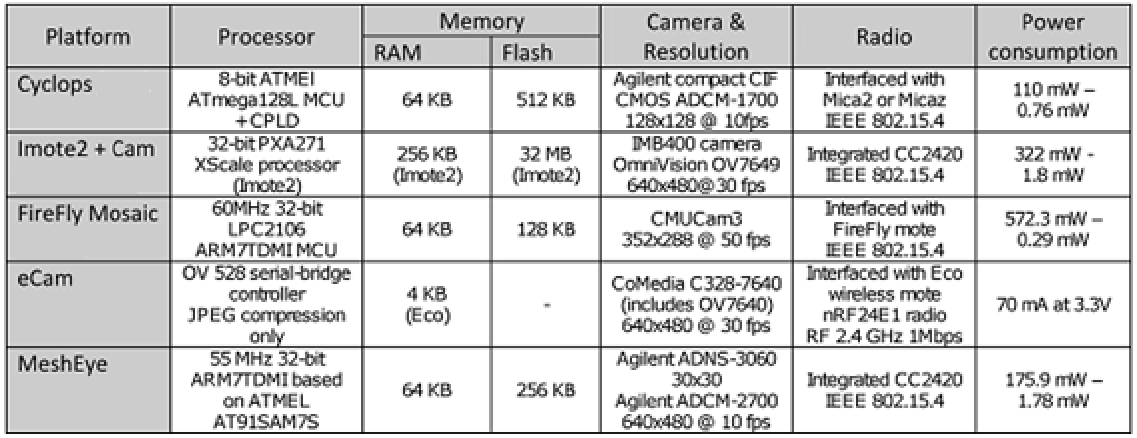
When more effective multimedia processing is required wireless motes may not be adequate. According to the requirements of the application (object detection to identification and tracking), appropriate resolution cameras and video processing algorithms are used within the mote. These type of devices form higher tiers of the network architecture and may be used as gateway between the underlying sensor network and the sink. Mote samples for this class are given in Table 2.2.

Table 2.1 Wireless Motes (Table Source [24])



	Wireless Mote	Microcontroller	Memory		Radio	Data Rate
			RAM	Flash Memory		
Lightweight-class	Mica2	ATmega128L (8 bit) 7.37 MHz	4 KB	512 KB	CC1000	38.4 Kbps
	Mica2Dot	ATmega128L (8 bit) 4 MHz	4 KB	512 KB	CC1000	38.4 Kbps
	MicaZ	ATmega128L (8 bit) 7.37 MHz	4 KB	512 KB	CC2420	250 Kbps
	FireFly	ATmega1281 (8 bit) 8 MHz	8 KB	128 KB	CC2420	250 Kbps
Intermediate-class	Tmote Sky	MSP430 F1611 (16 bit) 8 MHz	10 KB	48 KB	CC2420	250 Kbps
	TelosB	TI MSP430 (16 bit) 8 MHz	10 KB	1 MB	CC2420	250 Kbps
PDA-class	Imote2	PXA271 XScale (32 bit) 13 – 416 MHz	256 KB + 32MB SDRAM	32 MB	CC2420	250 Kbps
	Stargate	PXA255 XScale (32 bit) 400 MHz	64 MB	32 MB	CC2420 Bluetooth IEEE 802.11	250 Kbps 1 – 3 Mbps 1 – 11 Mbps

Table 2.2 Camera Motes (Table Source [24])



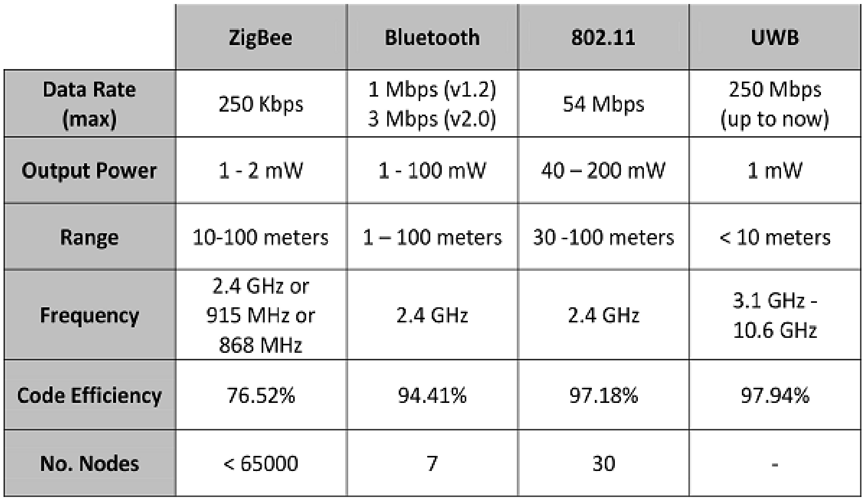
Platform	Processor	Memory		Camera & Resolution	Radio	Power consumption
		RAM	Flash			
Cyclops	8-bit ATMEI ATmega128L MCU +CPLD	64 KB	512 KB	Agilent compact CIF CMOS ADCM-1700 128x128 @ 10fps	Interfaced with Mica2 or Micaz IEEE 802.15.4	110 mW – 0.76 mW
Imote2 + Cam	32-bit PXA271 XScale processor (Imote2)	256 KB (Imote2)	32 MB (Imote2)	IMB400 camera OmniVision OV7649 640x480@30 fps	Integrated CC2420 IEEE 802.15.4	322 mW - 1.8 mW
FireFly Mosaic	60MHz 32-bit LPC2106 ARM7TDMI MCU	64 KB	128 KB	CMUCam3 352x288 @ 50 fps	Interfaced with FireFly mote IEEE 802.15.4	572.3 mW – 0.29 mW
eCam	OV 528 serial-bridge controller JPEG compression only	4 KB (Eco)	-	CoMedia C328-7640 (includes OV7640) 640x480 @ 30 fps	Interfaced with Eco wireless mote nRF24E1 radio RF 2.4 GHz 1Mbps	70 mA at 3.3V
MeshEye	55 MHz 32-bit ARM7TDMI based on ATMEL AT91SAM7S	64 KB	256 KB	Agilent ADNS-3060 30x30 Agilent ADCM-2700 640x480 @ 10 fps	Integrated CC2420 IEEE 802.15.4	175.9 mW – 1.78 mW

2.1.2 Communication

The development of a reliable and energy-efficient protocol stack is important for supporting various WSN applications [25]. Suitable communication standards and protocols should be selected according to the type of data to be transferred between sensor nodes. In terms of the physical layer, there are several technologies available for sensor-to-sensor communication; each has its own bandwidth, range, device number, and power consumption characteristics.

At the physical layer of the communication there exist mainly 4 communication standards with short range, low bandwidth and power usage attributes. The comparison of all four is given at Table 2.3. If multimedia streaming will not take place, ZigBee, which is based on the IEEE 802.15.4 standard, is the most common physical layer standard suitable for WSNs, based on its low power consumption and long range. It supports data rate of up to 250kbps, coding efficiency of 76.52%, supports more than 65000 nodes and effective within the range of 10-100 meters [23]. ZigBee standard is being used by most of WSN devices such as MICA-family, Tmote sky, and imote2 [25].

Table 2.3 Physical Layer Standards for WSNs (Table Source [24])



	ZigBee	Bluetooth	802.11	UWB
Data Rate (max)	250 Kbps	1 Mbps (v1.2) 3 Mbps (v2.0)	54 Mbps	250 Mbps (up to now)
Output Power	1 - 2 mW	1 - 100 mW	40 – 200 mW	1 mW
Range	10-100 meters	1 – 100 meters	30 -100 meters	< 10 meters
Frequency	2.4 GHz or 915 MHz or 868 MHz	2.4 GHz	2.4 GHz	3.1 GHz - 10.6 GHz
Code Efficiency	76.52%	94.41%	97.18%	97.94%
No. Nodes	< 65000	7	30	-

A ZigBee network consists of a coordinator device and several end devices. The coordinator selects a personal area network (PAN) ID that can be 16 or 64 bits. The other devices join this network via a coordinator device. A ZigBee module works either in transparent mode, transferring data coming to its pins directly to the air, or in packet-based mode, with which address-based packetized communication is performed. As ZigBee supports a data rate ≤250 Kbps, it is not feasible to use for dense data transfer applications, including multimedia streaming.

Bluetooth is another good candidate for communication in WSNs because of its low power consumption and low cost. It provides data rates of up to 1 Mbps and works consistently at a range of 10 m. Wi-Fi is another very popular wireless communication technology that is used specifically for local area networks (LANs). Wi-Fi is based on the IEEE 802.11 standard and supports much higher data rates at longer range than ZigBee; however, it consumes much more power than ZigBee, which makes it unsuitable for WMSNs.

In order to provide connectivity and reliability upper layer protocols can be implemented in sensor networks. The MAC layer is responsible for applying channel access policies, error control mechanisms and scheduling and buffer management. The goal is to enable error-free, reliable data transfer with minimum retransmissions while supporting quality of service (QoS) requirements [26]. Protocols at this layer mainly divided into two groups based on their channel access policy. Contention-Free protocols implement synchronization mechanism in order to gain media access. Time Division Multiple Access (TDMA) is a popular example for this group. On the other hand the second group, Contention-Based protocols, implements random access and do not require synchronization which makes them more power consuming approach. Carrier Sense Multiple Access (CSMA) is an example for this group. Although because of their simplicity, flexibility and scalability, those protocols are attractive for WSNs, multimedia applications which require strict QoS requirements over resource constrained WMSNs make use of those protocols infeasible in WMSNs [24].

There exist several routing protocols proposed for use in WSNs that must contend with a network’s natural challenges, including limited energy, random deployment, unknown node locations, and scalability. In addition, the routing protocol must also take the network architecture into consideration. A sensor network composed of clustered sensor nodes should be executed by a hierarchical routing protocol, rather than a plain routing protocol. In the present study the upper layer protocols and routing issues were not a primary focus. In order to achieve communication within the sensor network, only the most appropriate physical environment is selected.

Network and transport layers have their own protocols and approaches which are out of the scope of this work. Because of the existence of multimedia data, application layer consists of specific functions related with this data type, like source coding and image processing as well as traffic management and admission control functionalities. Application layer provides necessary operating system and software library support for the implementation of further data processing operations.

2.2 Object Detection and Classification

For automated video surveillance applications, image processing is one of the main challenging areas that focus on detection and identification of a threat from a given video stream. Moving object detection is the primary step and claims critical consideration in motion analysis and recognition system. Failure in this segment will cause the system to malfunction or operates inaccurately [17]. In order to provide intelligence and automaticity to the surveillance system, this part has crucial importance. Success of the latter steps, especially classification part is strongly depended on the quality of the extracted object.

There are several approaches in order to separate the threat or foreground from the static background. Frame differencing, optical flow and background subtraction are three important techniques which are used to extract the region of interest from the video stream. After retrieving region of interest further processing is required to get a clear view of the threat. For that purpose noise in the extracted foreground should be cleared and then object extraction and classification should be performed to handle the object.

In this section it will be focused on the background subtraction technique, noise elimination algorithms and object extraction techniques as sub-parts of object detection and extraction phase.

2.2.1 Background Subtraction

As video is the most widely used type of multimedia data for surveillance systems, the literature concerning the most commonly used methods of object extraction and identification from video was reviewed. The first step in identifying a threat based on video data is to extract the region of the threat from a video still image. Classification accuracy is strongly associated with the success of this phase [17]. One of the most popular methods used for region of interest (RoI) extraction is background subtraction (BS) and there are several approaches to BS. The simplest BS method, shown in (2.1), is calculating the difference in the pixel values between the current frame and the reference frame, which represents the background, and then applying a threshold to the result in order to detect the non-background areas [27];

$$Foreground_{(x,y)} = \begin{cases} 1 & d\left(I_{(x,y,t)},\ B_s\right) > \tau \\ 0 & Otherwise \end{cases} \tag{2.1}$$

where *I(x,y,t)* is the value of a pixel at time *t* and B_s is the value of the same pixel in the background frame.

According to this BS method, the background is considered as static, whereas especially in outdoor surveillance applications the background cannot be assumed to be constant. False positives can be induced by illumination changes, animated objects, or camera jitter. On the other hand, false negatives can also occur when a moving object is a color similar as objects in the background (the so-called camouflage effect) [28]. In order to provide a more adaptive background, statistical approaches are used. One method is to apply Gaussian average to update the background statistically; each background pixel is modeled with a probability density function (PDF) learned over a series of training frames [28]. This model is formulated as;

$$\begin{gathered} \mu_t=\rho\ I_t+(1-\rho)\ \mu_{(t-1)} \\ \sigma^2{}_t=\rho d^2+(1-\rho)\sigma^2{}_{t-1} \\ d=|I_t-\mu_t| \\ |I_t-\mu_t|>k\sigma_t\ (x,y), \end{gathered} \tag{2.2}$$

where μ_t is the mean and σ is the covariance values for each pixel. Those values are updated recursively. The ρ value is a weight constant. The covariance is updated according to the distance d between the mean value and the current value. The variable k is the threshold value used to determine if a pixel is in the foreground or background.

One of the most preferred background modeling methods is multimodal background modeling, which deals with multiple background objects at the same location at different time frames [29]. A pixel is modeled with a weighted combination of several PDFs rather than a single PDF and, as such, is known as the Mixture of Gaussians (MoG) model. In practice, the number of PDFs is set between 3 and 5. Simple methods, such as Gaussian average, offer acceptable accuracy while achieving a high frame rate and having limited memory requirements; however, more complex methods, such as the Mixture of Gaussians, display very good modeling accuracy [30].

2.2.2 Foreground Process

After RoI extraction, noise reduction and clutter elimination should be performed in order to clear unwanted structures. Furthermore, extracted foreground objects may also require post processing in order to make them sufficiently clear and sharp. The primary purpose of this post processing stage is to probe the image with a structuring element and to quantify the manner in which the structuring element fits (or does not fit) within the image. For that purpose the image is first converted to a binary image, and the primary morphological operations erosion-$\ominus$ and dilation-$\oplus$ are applied to this binary image [31].

Dilation is the morphological transformation which combines two sets using vector addition of set elements [31]. Set A is considered as the image undergoing analysis and set B is the structuring element. Following operation is an example for dilation.

A = {(0,1),(1,1),(2,1),(2,2),(3,0)} and B= {(0,0), (0, 1)}

$$A\oplus B=\{c\in E^n | c=a+b\ \ ,a\in A\ ,b\in B\}$$

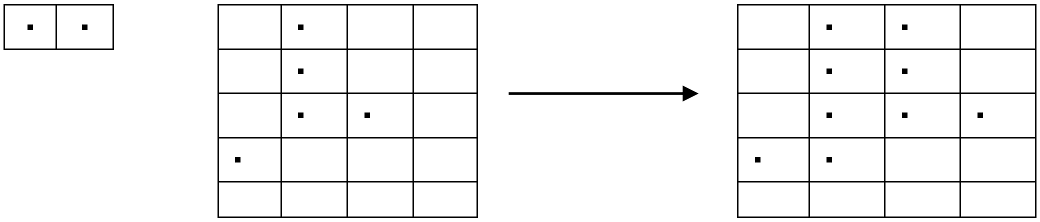


Figure 2.4 Dilation operation on image

Erosion is the morphological dual to dilation [31]. Like dilation it is combination of two sets but this time using vector subtraction. Following operation is an example of erosion:

A = {(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5),(2, 1), (3, 1), (4, 1), (5, 1),}

B = {(0, 0), (0, 1)}

$$A\ominus B=\{x\in E^n | x+b\ \ \in A\ \ \forall b\in B\}$$

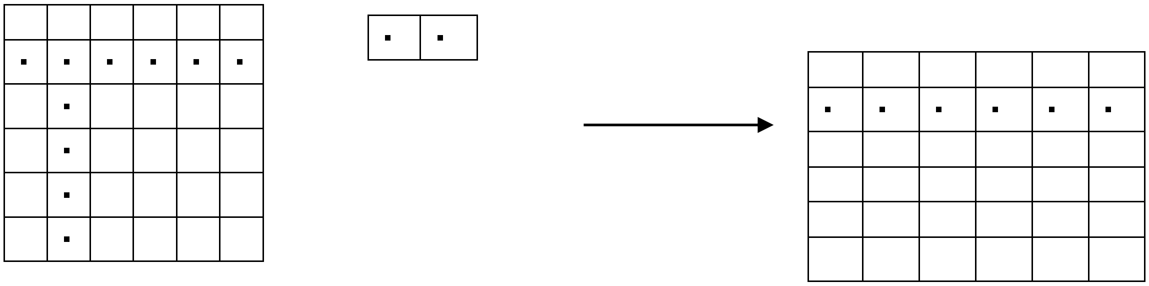


Figure 2.5 Erosion operation on image

Those two operations are applied one another iteratively. The result of iteratively applied dilations and erosions is an elimination of specific image detail smaller than the structuring element without the global geometric distortion of unsuppressed features [31]. Two important operations that are generated by the application of the basic erosion and dilation is opening and closing. Opening an image with a disk structuring element smooths the contour, breaks narrow isthmuses, and eliminates small islands and sharp peaks or capes. Closing an image with a disk structuring element smooths the contours, fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps on the contours.

The opening of image B by structuring element K is denoted by B o K and closing is denoted by B • K. The definitions of those operations are;

$$\begin{gathered} B \circ K = (B \ominus K) \oplus K \\ B \bullet K = (B \oplus K) \ominus K \end{gathered} \tag{2.3}$$

Before applying such operations, salt and pepper-type noises should be cleared from the foreground. Salt and pepper-type noises are usually caused by malfunctioning pixels in the camera's sensors. In the present study the source of salt and pepper-type noise was the residual areas following background subtraction. The median filter is among the most popular filters for removing this type of noise and, moreover, it is computationally very efficient [32]. The median filter is used to replace the pixel value with the median of the neighboring pixel values in its window. An example of application of median filter on 3 x 3 window is given in Fig. 2.6. When median filter is applied for the cell at the center, 97 value is replaced by the median value of all window (0, 2, 3, 3, 4, 6, 10, 15, 97) which is 4.

6	2	0
3	97	4
19	3	10

$$\xrightarrow[\text{Filter}]{}$$

*	*	*
*	4	*
*	*	*

Figure 2.6 Median filtering

2.2.3 Object Extraction

Following BS and post processing, the cleaned foreground image should be segmented in order to extract each object. Each object's bounding box (BB) is extracted via segmentation. There exist several image segmentation techniques that group pixels according to their similarity, based on color, intensity, and texture [33]. One of the most effective methods for image segmentation is connected component analysis (or connected component labeling), which detects connected regions in binary images. For a binary image, represented as an array of d-dimensional pixels or image elements, connected component labeling is the process of assigning labels to the BLACK image elements in such a way that adjacent BLACK image elements are assigned the same label [34]. Here, “adjacent” may mean 4-adjacent or 8-adjacent shown in Fig.2.7 Connected-component labeling can be characterized as a transformation of a binary input image, B, into a symbolic image, S, such that

(1) All image elements that have value WHITE will remain so in S; and,

(2) Every maximal connected subset of BLACK image elements in B is labeled by a distinct positive integer in S [34].

	N	
W	*	E
	S	

NW	N	NE
W	*	E
SW	S	SE

Figure 2.7 Pixel 4/8 adjacency

There are several algorithms to analyze the connected regions. Some recursive algorithms are based on the assumption that whole image can fit into memory. Other algorithms work on large images and process row by row. The classical row by row algorithm performs two passes on the image. At the first pass, equivalences between pixels are recorded and temporary labels are assigned to the pixels. At the second pass each temporary label is replaced by the label of its equivalence class

Edge detection generally refers to a group of image segmentation techniques that transform an image to an edge image based on changes in gray tones in the image [33]. There are 3 (three) main edge detection techniques referenced in the literature: Roberts, Prewitt, and Sobel. All 3 techniques detect gradient changes by calculating gray level differences between neighboring pixels. The thresholding approach is another common segmentation technique in which an image is partitioned based on ≥1 threshold values. This method is simply to use for partitioning an image to background and foreground areas. Furthermore, this method can be used to extract multiple objects using multiple threshold values.

2.2.4 Feature Extraction

In general, a feature can be considered as a unique subset of data differentiated from a larger body of data that can be used to identify an object. For computer vision features are used to identify objects in the foreground of a digital image. The most distinctive features of foreground object types, such as human, vehicle, and vegetation, should be determined and such objects in the foreground should be extracted prior to classification. Features can be simple structures, such as a point, corner, or edge, as well as more complicated structures, such as a texture, blob or object. Features are interesting part of the images that are used as starting point for many computer vision algorithms and content based image retrieval systems.

There are important requirements for feature points to have a better correspondence for matching [35]:

- Distinctiveness/informativeness: The intensity patterns underlying the detected features should show a lot of variation, such that features can be distinguished and matched.
- Repeatability: Given two images of the same object or scene, taken under different viewing conditions, a high percentage of the features detected on the scene part visible in both images should be found in both images.
- Locality: The features should be local, so as to reduce the probability of occlusion and to allow simple model approximations of the geometric and photometric deformations between two images taken under different viewing conditions.

- Quantity: The number of detected features should be sufficiently large, such that a reasonable number of features are detected even on small objects.
- Accuracy: The detected features should be accurately localized, both in image location, as with respect to scale and possibly shape.
- Efficiency: Preferably, the detection of features in a new image should allow for time-critical applications

Features can be categorized as shape-based features, such as aspect (width/height) ratio and shape complexity (perimeter2/area), texture-based features, such as Gabor features, and motion-based features such as speed [35]. As such features like aspect, compactness, and speed can be calculated during object extraction without any additional process, they play a significant role in energy efficiency and real-time object identification. There are several studies that have been focused on feature detection for image processing applications. Those studies are grouped according to their area of interest in [35]. Important group of studies are;

- Contour Curvature Based Methods: Applied to line drawings and focus was especially on the accuracy of point localization. Extracting points along the contour with high curvature is one of the strategies of those methods.
- Intensity Based Methods: Based on first- and second-order gray-value derivatives. There exist a few approaches in this group. Hessian-based approaches explore the determinant of Hessian matrix in order to extract blob-like structures. Gradient based approaches; one of the most famous is Harris corner detector, returns points at the local maxima of a directional variance measure.
- Color Based Methods: Proposed approaches based on color are simple extensions of methods based on the intensity change. Color gradients are usually used to enhance or to validate the intensity change so as to increase the stability of the feature detectors.

2.2.4.1 SIFT- scale invariant feature transformation

Most object features are affected by illumination changes, scaling, camera position/angle, and an object's position/orientation in three-dimensional space. To overcome this problem, researchers have been working to develop more robust feature extraction methods. One such method is scale invariant feature transformation (SIFT), as proposed by Lowe [36]. The aim of the algorithm is extraction of features which are invariant at light or scale changes and noisy environments and performing a reliable recognition based on those features.

This method transforms an image into local feature vectors each of which is invariant to translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection, via use of a staged filtering technique. Before feature detection a scale space for the input image is constructed by using Gaussian kernel function. This enables handling image structure at different scales. For a given image *f(x,y)* its Gaussian scale space representation is *L(x,y,t)* is convolution of the image with Gaussian kernel;

$$\begin{gathered} L(x,y,t)=g(x,y,t) \cdot f(x,y) \\ g(x,y,t)=(1/(2\,\pi\, t))e^{-(x2+y2)/2t} \end{gathered} \tag{2.4}$$

where *t* is scale parameter. Localization of the key by looking for locations that are maxima or minima of a difference-of-Gaussian function brings rotation invariance. Maxima and minima of this scale-space function are determined by comparing each pixel in the pyramid to its neighbors. Low contrast keys are discarded during this process.

After localization of the keys, the descriptors of those keys are calculated based on orientation histograms by using 4x4 pixel neighborhoods with 8 orientation bins for each (total 128 dimensional vector). The resultant histogram is shown below in Fig.2.8.

Matching with the keys of different images is identification of most similar keys for high dimensional vectors which has high complexity. In order to simplify it [14] proposes to use best bin search method based on a modification of k-d tree algorithm [15]. The keys produced at larger scales are weighted twice the weight of the lower scales in order to increase the efficiency of the algorithm.

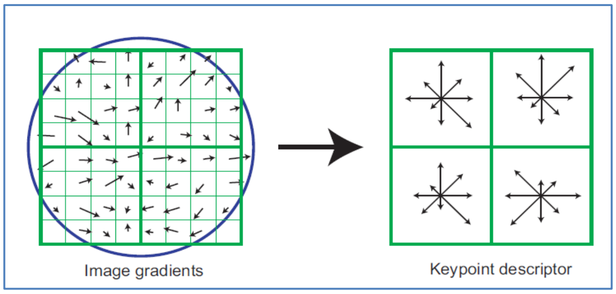


Figure 2.8 Feature descriptors (Figure Source [36])

2.2.4.2 SURF- Speeded up robust features

Speeded up robust features (SURF) is a similar technique that was proposed by Bay et al. [37] and is claimed to be more efficient than SIFT for feature extraction. SURF applies Hessian matrix approximation to an integral image [38] rather than an original image to detect feature points. The integral image at location *(x,y)* simply represents the sum of all pixels in the input image *I* within a rectangular region formed by the origin and *x*.

The interest point detection in SURF is based on Hessian matrix approximation because of its performance in accuracy. The points where determinant of the Hessian is maximum are chosen as interest points. Given a point *x=(x,y)* on image *I* the Hessian Matrix *H(x,σ)* in *x* at scale *σ* is ;

$$H(x,\sigma)=\begin{bmatrix} L(x,\sigma)_{xx} & L(x,\sigma)_{xy} \\ L(x,\sigma)_{yx} & L(x,\sigma)_{yy} \end{bmatrix} \qquad (2.5)$$

where $L(x,\sigma)_{nn}$ are convolution of the Gaussian second order derivative of gray scale image.

The description of the interest point describes the distribution of intensity content within the interest point neighborhood, similar to gradient information extracted by SIFT. Rather than the gradient, first order Haar wavelet distribution in x and y direction is used. A neighborhood of size 20x20 is taken around the key point. It is divided into 4x4 sub regions. For each sub region, horizontal and vertical wavelet responses are taken. This when represented as a vector gives SURF feature descriptor with total 64 dimensions.

2.2.4.3 HOG- Histograms of Oriented Gradient

Histograms of Oriented Gradient (HOG) descriptors are also scale invariant features that were originally proposed for detection of humans in images [39]. The overview of the method is given in Fig.2.9. The HOG method is based on the notion that an object can be sufficiently characterized according to the distribution of local intensity gradients or edge directions. The image is divided into blocks and histogram of gradient directions are extracted and combined for each block. In order to minimize the effect of illumination changes, shadows etc. it is proposed to normalize contrast prior to processing.

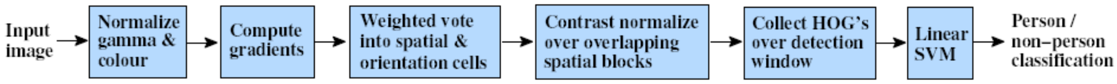

Input image
Normalize gamma & colour
Compute gradients
Weighted vote into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Collect HOG's over detection window
Linear SVM
Person / non−person classification

Figure 2.9 HOG overview (Figure Source [39])

The primary disadvantage of all these invariant features is that extracting them from an image, and storing and using the extracted data for classification is not processing, power, or memory cost effective.

2.3 Classification

The last stage of the object identification process is detection of the category of the extracted foreground object based on its features. Features that are extracted for each object (size, shape, interest points, texture…) are means to identify the classes of objects. The classification is mostly based on the predetermined classes and the most appropriate class is tried to be assigned to the detected object. Like all other phases this operation is considered as an in-network task which should be handled with the well-known resource constraints. Integration of this classification information with further info like localization and tracking is the fundamental step of event generation which is the main goal of automatic surveillance application.

A suitable classification system and a sufficient number of training samples are prerequisites for accurate classification [40]. Accordingly, in order to achieve accurate classification, the classification system should be trained using a sufficient number of samples for each target object class, so that the extracted features will be representative of each object class. In this way the classification system is able to identify an object as the most appropriate class by comparing the features of an object one seeks to classify to the training data set.

Summarizing the definitions, main steps of the classification are [41];

- Definition of classification classes: Depending on the objective and the characteristics of the image data, the classes should be clearly defined.
- Selection of features: Features to discriminate between classes should be established.
- Sampling of training data: Training data should be sampled in order to determine appropriate decision rules. Supervised or unsupervised technique should be selected based on the training dataset.
- Estimation of universal statistics: Various classification techniques will be compared in order to find an appropriate decision rule.
- Classification: Depending on the decision rule pixels are classified in a single class.

- Verification: The classified results should be checked and verified for their accuracy and reliability.

In general, image classification approaches can be grouped as supervised and unsupervised, or parametric and nonparametric, or hard and soft (fuzzy) classification, or per-pixel, per-object, and per-field. Table 2.4 provides brief descriptions of these categories [40]. For the surveillance applications selection of the approach according to those categories is important because of the resource constraints and real time needs. Need for training data set and descriptor calculation bring complexity and require processing power and extra storage capabilities. As a result keeping the classification phase simple and effective is important for real time video surveillance applications.

As for all steps of the object identification process, the method selected for feature matching and classification should meet WMSN constraints. In this section several classification methods that can be adapted to the proposed architecture are described, such as the Naïve Bayes Classifier, k-Nearest Neighbors (k-NN) algorithm, Support Vector Machine (SVM) [42] and Bag of Words (BoW).

2.3.1 Naïve Bayes Classifier

Naïve Bayes is a simple statistical supervised classification which is based on the idea that input features for classification are conditionally independent of each other. In other words, the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. One of the important advantage of this classifier is it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. The classifier is based on the Bayes theorem which is;

$$P(C/F_1,F_2,F_3,\ldots F_n) = \frac{P(C)P(F_1,F_2,F_3,\ldots F_n|C)}{P(F_1,F_2,F_3,\ldots F_n)} \tag{2.6}$$

Table 2.4 Classification Approaches (Table Source [40])

Criteria	Categories	Characteristics	Example
Use of training samples	Supervised	The signatures generated from the training samples are then used to train the classifier to classify the spectral data into a thematic map	Maximum likelihood, minimum distance, artificial neural network, decision tree classifier
	Unsupervised	Clustering-based algorithms are used to partition the spectral image into a number of spectral classes based on the statistical information inherent in the image. No prior definitions of the classes are used. The analyst is responsible for labeling and merging the spectral classes into meaningful classes.	K-means clustering algorithm
Use of parameters (mean vector, covariance Matrix)	Parametric	Gaussian distribution is assumed. The parameters (e.g. mean vector and covariance matrix) are often generated from training samples	Maximum likelihood, linear discriminant analysis
	Non-parametric	No assumption about the data is required. Classifiers do not employ statistical parameters to calculate class separation and are especially suitable for incorporation of non-remote-sensing data into a classification procedure	Artificial neural network, decision tree classifier, evidential reasoning, support vector machine, expert system
Pixel information	Per-pixel	Traditional classifiers typically develop a signature by combining the spectra of all training-set pixels from a given feature. The resulting signature contains the contributions of all materials present in the training-set pixels, ignoring the mixed pixel problems	Most of the classifiers, such as maximum likelihood, minimum distance, artificial neural network, decision tree, and support vector machine
	Object-oriented classifiers	Image segmentation merges pixels into objects and classification is conducted based on the objects, instead of an individual pixel.	eCognition.

where C is the class variable, $F_1,F_2,F_3,\ldots F_n$ are feature variables, $P(C| F_1,F_2,F_3,\ldots F_n)$ is the probability of class C under the existence of $F_1,F_2,F_3,\ldots F_n$ features and $P(F_1,F_2,F_3,\ldots F_n)$ is evidence and constant when the values of the features are known. The numerator of the equation is equivalent to $P(C,F_1,F_2,\ldots F_n)$. The translation of the formula is *posterior=(prior x likelihood)/evidence.*

Use of the NB classifier for object recognition is the problem of finding most probable class according to the features extracted from the detected object and categorized according to the learned model [20]. The steps can be summarized as;

- Training data set is used to extract features and features are clustered and labeled with its corresponding class.
- Features of the detected object are extracted.
- For each feature *f*, the most probable cluster of features k_f from the training data set is selected. This selection is based on the distance D between the cluster and the feature.

2.3.2 k-NN Classifier

k-NN is non-parametric (does not make any assumption on the distribution of data) and lazy learning (does not use training data points for generalization) method for classification. It is simple and first choice for classification especially when there is little or no prior knowledge. Classification, as shown in Fig.2.10, is simply a matter of locating the nearest neighbor in instance space and labeling the unknown instance with the same class label as that of the located (known) neighbor. *k* is the number of neighbors to be examined and the algorithm lets the majority vote decide the outcome of the class labeling.

The simplest neighborhood search method is measuring the Euclidian distance between the test sample and the prior sample. For input x_i with p features $(x_{i1},x_{i2}\ldots x_{ip})$ and a prior sample x_j with features $(x_{j1},x_{j2}\ldots x_{jp})$, the distance between x_i and x_j ;

$$d(x_i,x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots (x_{ip} - x_{jp})^2} \quad (2.7)$$

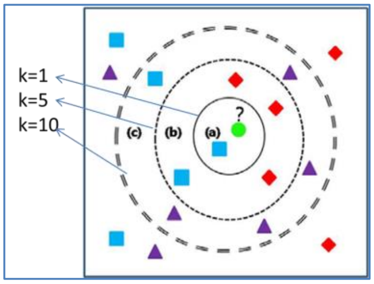


Figure 2.10 KNN classification

2.3.3 SVM-Support Vector Machine

Another popular and effective method is Support Vector Machine (SVM) [42], a supervised classification method. Using the machine a linear decision surface is constructed, by the help of a so called Kernel Function, from a non-linearly mapped feature space. In this decision surface a separating hyper-plane which has the maximum distance to the data points is selected. The maximum margin to data points means less generalization error. An example of maximum-margin hyper plane is shown in Fig.2.11(a) [43]. In the figure H_1 plane does not separate the data. Meanwhile H_2 does but plane is too close to the data points which results with a bad separation. Finally H_3 is able to separate the data with maximum margins. The H_3 is defined as linear decision function or decision boundary. Given a set of training data in the form of $(X_{11},X_{21})\ldots(X_{1i},X_{2i})$ the separating hyper plane is the set of points which meet following rule;

$$w{\cdot}x - b = 0 \quad (2.8)$$

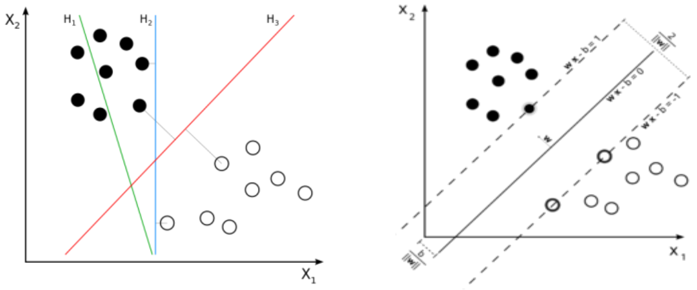
where w is the normal vector to hyper-plane, and $b/||w||$ is the offset of the hyper-plane from origin along w and dot product $w.x$ is the projection of the vector x along w. For linearly separable data the separating hyper-plane is just between the two hyper-planes which separates the two classes of data with the largest distance as shown in Fig.2.11(b) [43]. Those two hyper-planes are composed of support vectors of the training data. In two classes case the support vectors can be found as;

$$\begin{aligned} w \cdot x - b &= 1 \\ w \cdot x - b &= -1 \end{aligned} \qquad (2.9)$$

If we think our classes as “+ ” and “–” then for “x_+” and “x_-” samples we conclude that;

$$\begin{aligned} w \cdot x_+ - b &\geq 1 \\ w \cdot x_- - b &\leq -1 \end{aligned} \qquad (2.10)$$

The real contribution of SVM is when the input vectors are linearly inseparable. In such a case the transformation function called kernel function is applied on the input vectors in order to put them into a linearly separable higher dimensional feature space. This is also called “Kernel Trick”[44]. Some popular kernel functions are given in Table 2.5 below and mapping is demonstrated in Fig.2.12.



(a)

(b)

Figure 2.11 a) Hyper-plane selection for linearly separable data, b) Margins of linearly separable two classes of data (Figure Source [43])

Table 2.5 Popular Kernel Functions

Type of Kernel	**Inner Product kernel**
Polynomial Kernel	$K(x,x_i)=(x^T \cdot x_i + C)^d$
Gaussian Kernel	$e^{\frac{1}{2\sigma^2}\|\|x-xi\|\|^2}$
Sigmoid Kernel	$tanh(\eta x.x_i + \theta)$

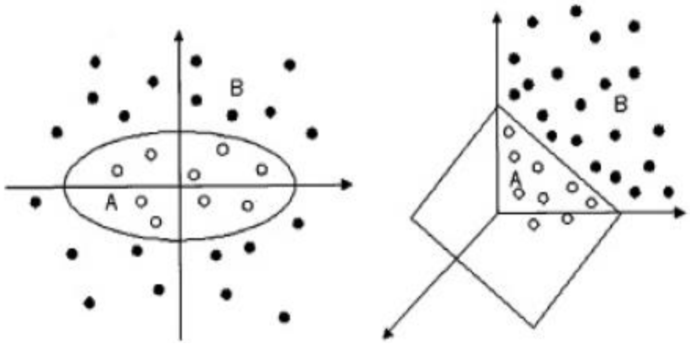
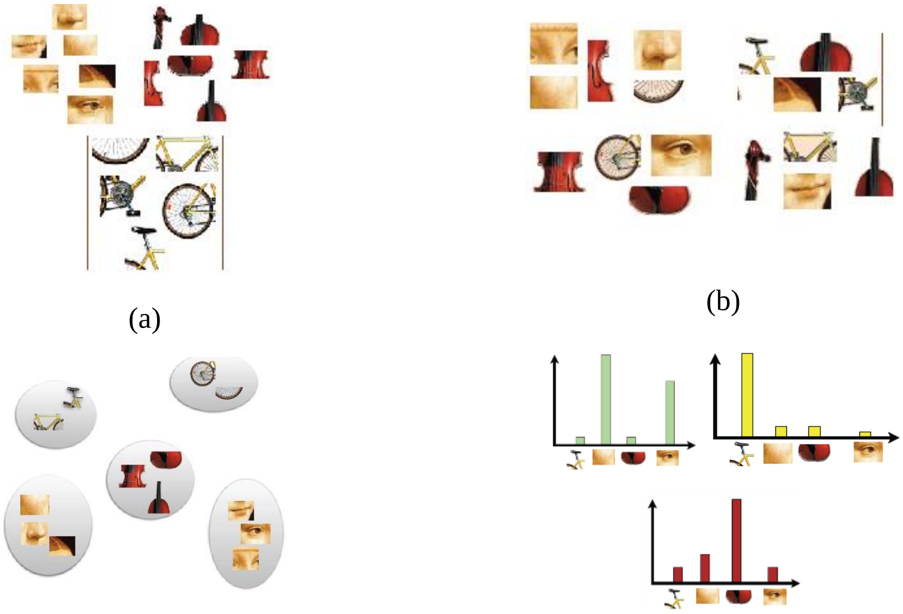


Figure 2.12 Mapping linearly inseparable data to linearly separable feature space (Figure Source [44])

2.3.4 BoW-Bag of Words

Bag of Words (BoW) is a popular approach to representing an image as groups of features. Rather than using similarities between features, the histogram of features is used to classify an object. In computer vision the process starts with extraction of features. For that purpose algorithms like SURF, SIFT or HOG can be utilized as described above. When the descriptors of the extracted features are available, those descriptors are clustered. In this way a visual vocabulary which consists of codewords is constructed based on clustered descriptor. The frequencies of those codewords are calculated and such classifiers as SVM are applied to the codewords histogram to identify the most relevant object class. The stages of the method are demonstrated in

Fig.2.13 [45]. This method is also a candidate for use with the proposed architecture in order to utilize local features during the classification process.



(a)

(b)

(c)

(d)

Figure 2.13 Bow Stages a) Extraction of visual features, b)Constructing visual vocabulary, c) Calculating frequencies d)Representing images as histogram of words. (Figure Source [45])

2.4 Previous Studies

Moving object detection, classification, and tracking using multimedia data have been extensively studied. Many studies have been performed based on resource-rich wired surveillance systems. Such systems are capable of producing high-resolution images, and can also employ complex algorithms and methods due to their powerful processors and unlimited power sources. Besides such resource-rich applications, the primarily focus here is on resource-constrained platforms and WSNs. Development of sensor node hardware capable of both sensing and processing data is one of the primary goals of surveillance network research.

Lin et al. [19] propose surveillance system for street environment which deals with three classes; vehicles (including cars and trucks), motorcycles (including bicycles), and pedestrians. Three main functions of the proposed system are detection and tracking, recognition and classification and event summarization. For moving object detection, background subtraction is chosen. At the tracking part, it is assumed that the movement of the object is uniform so that the location of the object at the next frame can be predicted based on the displacement of the target between two consecutive frames. Features selected for the classification of the objects are height/width ratio and the walking rhythm. The flowchart of the whole study is given in Fig.2.12. The walking rhythm is chosen in order to solve the false positive issue and occlusion problems. The system is tested by using few street videos which have different illumination conditions. The success rate under consistent lighting conditions is about %98 and the overall success rates for pedestrian, motorcycle and vehicle are 87.5%, 98.4% and 94.7%, respectively.

Adolph and Reisslein [46] provide a detailed survey and comparison of smart sensor node research. In their study they identify 3 core requirements for wireless video sensor platforms (WVSP); power consumption, throughput and cost. They define those requirements in detail and according to those main requirements they select WVSPs that meet their criteria as much as possible. Selected platforms are divided into three main categories as; General Purpose, Heavily Coupled, and Externally Dependent architectures. After giving an overview of each category, they evaluate platforms of each category separately. Platforms are assessed based on their processor speed, power usage and modes, memory/storage modules, I/O interfaces, Radio attributes, Imaging specifications, operating systems, and cost. At last a flexibility rating is given to each of the platform within the evaluated category.

Kulkarni et al. [11] design and implement a three-tier network of heterogeneous wireless nodes and cameras, refer to as SensEye. They use low-resolution cameras (Cyclops) [47] and low-power motes at the first tier. In the second tier they include StarGate [48]

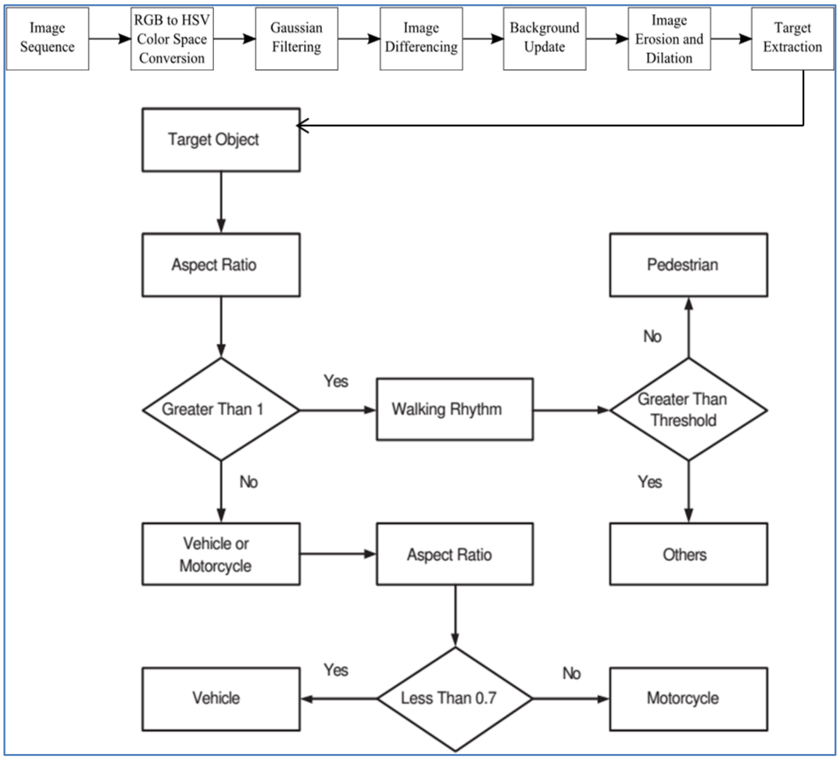


Figure 2.14 Flowchart of detection and classification (Figure Source [19])

nodes with web-cams. In the third layer they use high-resolution pan-tilt-zoom (PTZ) cameras connected to workstations. Object detection is performed at tier-1 and then tier-2 is activated for further processing, including localization, recognition, and tracking. Architecture heterogeneity is the result of the specifications of the cameras used in each layer. They propose to activate camera and sensor nodes periodically, so as to prevent continuous energy consumption. They do not benefit from scalar sensors for the detection of objects and initialization of cameras, which can improve energy consumption and performance. Their object recognition process is not among their primary concerns, as they report that any recognition algorithm can be employed. They use a face recognition algorithm and simple color-based heuristic to match objects to the image database. The accuracy of those methods for object recognition is not studied in detail. Moreover, the role of shape-based features and robust features in object recognition is not sufficiently tested.

Borgano et al. [16] grouped video surveillance applications into two categories in their study. Those are; scene dependent applications which requires a training phase and concentrate on specific object types and scene-independent solutions where algorithms are designed to be unaffected by variations in camera view, camera angle, object position and orientation. Scene-independent multi-class recognition should take into account at the design or training stage all possible classes of interest because adding new models or new training sets for different camera views would largely increase the configuration time. According to their observation 9 classes are detected as most commonly present in video surveillance applications. Those are package, person, bicycle, motorcycle, group of people, crowd, car, van and lorry or bus. Features that are used to identify those classes are height, width, area, dispersedness, border distance to centroid, speed and averages and variances of those features. Classification according to the features is performed according to three different models. Those are; rule based classifiers which are running using fixed predefined parameters, k-means classification which is based on measuring the Euclidean distance between the centers of the clusters and the feature vector of the observed object and AdaBoost which is a fast and strong classifier which is linear combination of weak classifiers. According to the scene independent classification performed via those methods has poor performance mainly due to the wide variation in the class features caused by scene change.

The proposed solution in the study [16] is a scene dependent one which requires training the system separately for each camera view (by using data acquired by the video surveillance system during a fixed period of time), so that it can adapt to the different characteristics of the scene. In the classification process, a similarity measure between the observed object instance and the 32 labeled samples is calculated to assign the object to a specific class. Scene-independent parameters plus position, horizontal and vertical speed features are used for classification. The training processes showed that the most significant features are position, height, width, aspect ratio, area, area ratio and absolute, horizontal and vertical speed. The results of the scene dependent learning based solution are compared with the ones achieved by the scene-independent solutions. It is observed that the learning-based method achieves better results for all classes, in particular for motorbikes and groups of people.

Lipton et al. [49] proposes a simple method based on temporal differencing and image template matching. They avoid use probabilistic approaches like Kalman filtering, in order to decrease the complexity. Unlike most surveillance applications the system detects moving objects, in fact moving regions, by using temporal frame differencing. The target object is classified based on two key elements. The first one is the dispersedness which is based on the perimeter and the area. The system distinguishes the target into two classes; human which has more complex shape so that larger dispersedness and vehicle. The second key element is the temporal consistency. The consequent motion regions are matched so that the statistics of a particular potential target can be built up over a period of time until a decision can be made about its correct classification. Furthermore, transient motion regions such as trees blowing in the wind will be thrown away. A simple application Maximum Likelihood Estimation is used for classification. If the target persists for time, the peak of the classification histogram during that time is used to classify the target. For tracking, detected motion regions are correlated with the pre-classified regions which are used as training templates. After the best correlation has been found the template is updated to ensure that the current template accurately represents the new image of the object. The result of the study shows that 86% of the vehicles and 82% of the humans are correctly classified with this method.

Brown [21] propose a method which is not limited to certain camera viewpoint directions (far field), is not linear/planar, nor does it require objects moving at a constant velocity. The classifier determines if the track is a person or a vehicle. The system is composed of three phases. At the first phase a straightforward classification is performed by using k-NN algorithm and following features; compactness, variation in compactness in time, fitted ellipse major/minor axis ratio, fitted ellipse near horizontal/near vertical axis ratio, velocity and direction. Normalization of the features according to the collected data is performed at the second phase. Following normalized features are added to the previous features; normalized major axis, normalized minor axis, normalized area. Performance of the whole system is compared with respect to normalized and un-normalized phases and different camera viewpoints. It is reported that the classification of individual frames (based on 24,309 frames) improved from 92% to 97% after normalization.

Jelii et al. [9] propose a multi-tier wireless video sensor network (WVSN) that uses a low-power wireless node. The proposed system simply uses a collection of PIR sensors in the lower tier that activate a camera sensor in the upper tier, according to a rule base. The camera sensor is connected to a workstation and sends alarms and images. They use ZigBee modules for communication. Their study focuses on energy consumption by the nodes and coverage area. The sensor nodes used are not intelligent. The images from the cameras are streamed to an upper layer station without processing, thereby increasing energy consumption and requiring operators to analyze the transferred real-time data.

Chen et al. [12] propose a video surveillance system consisting of many low-cost sensors and a few wireless video cameras. All the sensors are aware of their position. The nodes send event data to a sink station and the sink station triggers appropriate cameras. As in the study by Jelii et al. [9], the cameras stream video back to the sink station for further data analysis, which increase the amount of energy consumed by communication. Image/data processing and event/object recognition are not performed neither at the sensor nodes nor at the sink station, which also makes it necessary for multimedia data to be analyzed by human operators. The only outcome criterion in that study is the ratio of the number of monitored events to the total number of events.

Yasar et al. [13] propose an energy efficient object detection and image transmission approach for WMSN. In their study they perform a probabilistic threshold calculation on the captured image and fuse it with the threshold of background subtracted result. In this way they try to predict presence or absence of the object. According to the prediction result they send the image to the sink station. They perform simulations in order to evaluate the performance of their approach. No real world experiments are performed. Although their approach seems energy efficient, image transmission for detected objects which has negative effect on energy consumption. Besides, object categorization is not included in their study.

Kandhalu et al. [5] describe a smart camera sensor system based on the development of a sensor node referred to as DSPcam that has a CMOS camera, DSP/RISC-type processor, I/O interfaces, and a Wi-Fi module for connectivity. The sensor node is connected to a wireless sensor network using a serial interface. Although the system is capable of motion and object detection using frame differencing, they do not implement object or event recognition, as such, interpretation of all the events is performed by a human operator.

Damarla et al. [50] study the detection and identification of people and animals using non-imaging sensors, including acoustic, seismic, and ultrasonic transducers. They process the data obtained by the non-imaging sensors to extract event-based features and apply algorithms to detect people. They use scalar data only.

Clemsen et al. [51] devise an embedded platform capable of extracting information from a surveillance video stream in real time. They detect objects using a BS algorithm and Viola-Jones detector. In addition, they perform object tracking using the Kalman Filter method. Their proposed system is used in 2 real-world applications: vehicle detection on highways and license plate detection in urban traffic videos. Rather than a sensor network, their system works as a standalone video server that produces a labeled image to its users.

Chitnis et al. [52] design a WSN framework based on line sensor architecture. Unlike traditional video sensors that produce two-dimensional images, line sensors generate a one-dimensional image stream. Their goal is to increase the speed of image processing operations and decrease the required bandwidth, storage, memory, and power. A background is calculated using input line image averages and the foreground is discerned via subtracting every input image from a pre-calculated background. They transfer a report about an object as well as foreground image to the base station. Lastly, they extract the boundaries of a moving object. Object identification is not an ability included in the application and they do not use scalar sensors. Data communication in the network occurs directly between the sensor node and base station; the sensors do not communicate with one another. In summary, their system can be considered as a collection of independent sensor nodes that are managed by a workstation, rather than a WSN.

Oztarak et al. propose system architecture for classifying human, animal, vehicular objects in a WMSN [53]. Their system is designed to extract the minimum bounding rectangle (MBR) of a moving object in a captured frame using frame differencing. The system then calculates a membership value based on the width/height ratio of the MBR. Next, they recommend using rule-based classification to determine the category of a detected object. Their proposed methods rely heavily on assumptions. They do not perform any implementation related to their methods, sensor nodes, or network. Furthermore, their experiments are conducted using a simulated environment and specified conditions.

Sun et al. [14] propose border control system consist of multimedia sensors and scalar sensors. The system is composed of three layers. At the first layer there exist resource-constrained, low-power scalar sensors, which perform simple tasks such as taking seismic/vibration measurements and sending the information to data sink or processing hub. At the second layer, powerful and reliable multimedia sensors which can act as local processing hubs exist. Those are responsible for collecting data from scalar sensors, detecting possible intrusion according to the sensing reports as well as the local image/video information and reporting the detected results to the remote administrators. Third layer provides additional capabilities by using unmanned aerial vehicles (UAV) and robots. Those can be equipped with on board camera and sensor systems to provide additional coverage on demand basis. In order to reduce human involvement during the threat detection stage, two methods are proposed for detection in the study. First one is centralized method which requires compression of images locally and sending compressed data to remote processing center which has high computation capacity. The other one is distributed method where camera sensors perform detection locally.

Kim et al.[54] propose a robust and efficient multi-object recognition scheme that can be executed effectively on mobile devices. The object recognition is performed using local features from which the descriptors extracted by SURF. Before starting recognition the system is trained by using training set for each type of object. The interest points and their descriptors are extracted and after a statistical analysis on this data the representative points for each object type are selected. An interest point which has an enough number of similar interest points in terms of the SURF descriptor is considered as a representative point. Based on those representative points, it is calculated the threshold for each object that will be used during recognition. At the recognition stage, the interest points and their descriptors are extracted. These interest points are compared with the representative points of the objects in the training set. If the matching ratio for an object is higher than the object's threshold, then we consider the query image to have the object. The training phase (pre-processing) and recognition phase (query processing) are given in Fig.2.13

It is observed that most processing is performed during SURF descriptor extraction and matching. It is tried to decrease the number of comparisons by merging the representative points. The pre-processing stage is modified so that there merged interest points are extracted with different descriptors than their SURF descriptors. The new descriptors represent not a single feature value but a feature range covering all merged representative points. The tests are performed by using 4 types of objects (stop signs, motor bikes, yin-yang symbols and faces). 4 different approaches are tested; S1: basic algorithm, S2: basic algorithm with dynamic weights, S3: basic algorithm with merging, S4: basic algorithm with dynamic weights and merging. The basic method shows variations at recognition accuracy (76%–93%) for different object types. The accuracy is improved by using weighting. The merged features however increase the rate of false alarms.

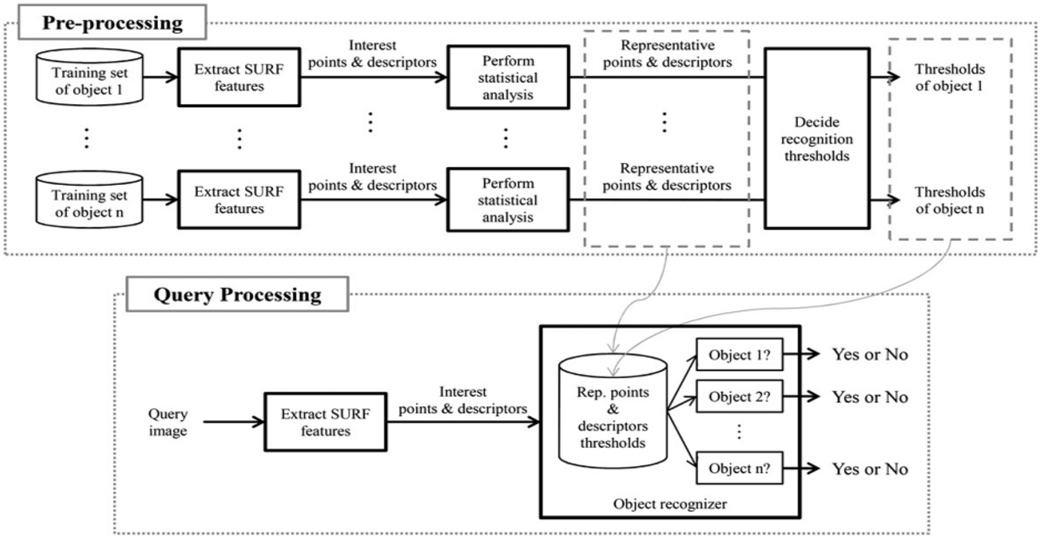

Pre-processing
Training set of object 1
Extract SURF features
Interest points & descriptors
Perform statistical analysis
Representative points & descriptors
Decide recognition thresholds
Thresholds of object 1
Training set of object n
Extract SURF features
Interest points & descriptors
Perform statistical analysis
Representative points & descriptors
Thresholds of object n
Query Processing
Query image
Extract SURF features
Interest points & descriptors
Rep. points & descriptors thresholds
Object 1?
Object 2?
Object n?
Yes or No
Yes or No
Yes or No
Object recognizer

Figure 2.15 SURF based surveillance flowchart (Figure Source [54])

CHAPTER 3

PROPOSED WMSN ARCHITECTURE

3.1 Architecture and Components

We designed a multi-tier automated surveillance system for outdoor applications, which is composed of wireless multimedia sensors and scalar sensors. The system will be developed to setup in to the target area without a reliable, hard-wired connection mechanism and permanent power supplies. It will continue to perform the desired actions with almost no human intervention. It is proposed to detect three types of objects by using the proposed system. Those are; people, vehicle and group of people.

This system is composed of 3 (three) layers. The first layer includes scalar sensors with acoustic, vibration, and motion sensing capability. This layer activates the second layer, which is composed of multimedia sensors with video processing capability. The concept information related with the sensed object is extracted and forwarded at this layer. The third layer which consists of a sink server is responsible for collecting all information form the surveillance network and sharing this information with the users of the system. The diagram of the architecture of this multi-tier system is given in Fig.3.1.

3.1.1 Scalar Sensor’s Layer

Scalar sensors at the first layer perform initial detection of an intruder, even if it is outside the view of the cameras. Passive infrared sensor-PIR, seismic, and acoustic sensors are potential instruments for that layer. A motion sensor, PIR, transforms the detection of motion into an electric signal. PIR sensors detect body heat in a range of

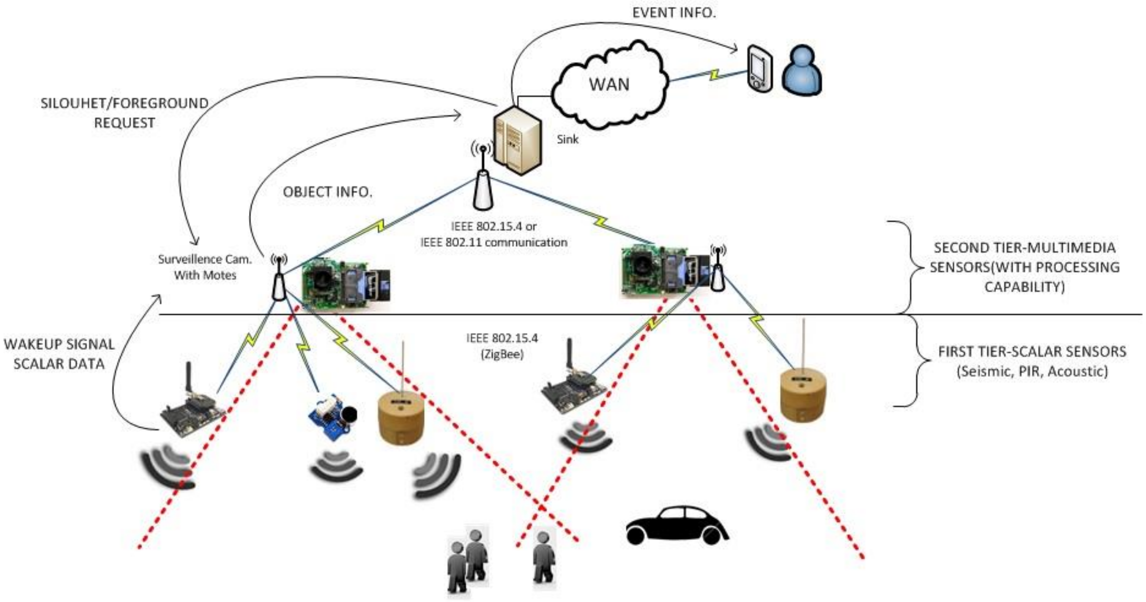

EVENT INFO.
WAN
SILOUHET/FOREGROUND REQUEST
Sink
OBJECT INFO.
IEEE 802.15.4 or IEEE 802.11 communication
Surveillence Cam. With Motes
SECOND TIER-MULTIMEDIA SENSORS(WITH PROCESSING CAPABILITY)
WAKEUP SIGNAL SCALAR DATA
IEEE 802.15.4 (ZigBee)
FIRST TIER-SCALAR SENSORS (Seismic, PIR, Acoustic)

Figure 3.1 The proposed WMSN architecture

15-25 meters. PIR sensors are mostly integrated within the cameras but standalone sensors that have IEEE 802.15.4 (ZigBee) capability-WiPIR also exist.

Seismic sensors pick up mechanical oscillations of the ground and convert them into electrical signals. It improves the security by detecting foot-borne intruders. Especially for perimeter surveillance, seismic interference is an important concern. However studies have shown that performance of those sensors are very prone to noise sources. So those sensors are proposed to be used with an algorithm for noise cancellation. On the other hand getting the raw signal and dealing with the semantic in the upper layer is still an approach. In our architecture we propose to use the seismic sensor in order to detect existence of vibration.

Acoustic sensors are able to measure the sound levels. The detection mechanism is the propagation of the sound waves though metal surfaces. Any change in the waves is converted to digital signals. Acoustic sensing is applied in various areas of security domain like content analysis, people tracking, vehicle classification, and gun shooter localization. Like the other scalar sensors we propose to use this device in order to detect existence of sound in the environment.

Such lower layer scalar sensors can contain lightweight motes in order to provide limited processing and communication capabilities. In this way it would be possible to provide semantic data to the upper layer.

3.1.2 Multimedia Sensors Layer

The second layer is composed of multimedia sensor nodes equipped with video cameras, a processor, and communication and storage modules. After detection of an intruder via scalar sensors at the first layer, a related multimedia sensor is activated. This is either done via I/O interfaces of the sensor node if the scalar devices are directly connected or via the IEEE 802.15.4/MAC communication link between the sensor and the node. In our proposed system scalar sensors are wired directly to the multimedia node. Those sensors produce simple high/low signals or scalar values that can be read from the multimedia sensors' I/O interfaces. This smart node includes a low or medium-resolution CMOS video camera with satisfactory field of view and depth of field values. This type of camera is chosen because of its lower energy consumption than CCD (charge-coupled device) cameras. Those multimedia sensors are located so that no blind area remains.

The application on each multimedia sensor node uses those input signals to activate the video camera and begin executing object extraction or classification operations. Scalar values, if available, can also be fused with multimedia data to facilitate object classification.

Those multimedia nodes also consist of microprocessor, memory storage and communication modules. Image processing operations that are required to extract the semantic information are performed in the node. Extracted semantic data is then forwarded towards the upper layer, sink, by means of communication modules. Nodes also run an operating system and necessary libraries in order to execute those image processing operations.

After capturing a frame, the multimedia sensor node begins image processing operations to extract and identify the objects in the frame. In order to perform object classification, the multimedia nodes train themselves using a sample image dataset, which includes a number of images of each object category that the sensor node is expected to classify. Prior to use of application, predetermined features of those sample images are calculated and stored. When a new object is extracted from video data its features are matched to the stored training data features and are evaluated based on the best-fitting category. Rather than forwarding raw multimedia data, the sensor node forwards the semantic information it extracts from the multimedia data.

3.1.3 Data Transfer

Multimedia nodes are clustered in order to forward their data. A number of nodes are connected to a cluster head and send their data to this cluster head via wireless interfaces. Serial radio frequency (RF) based communication between sensor nodes is used in order to limit power consumption. The sensor nodes forward classification results in text-based format. In this way the quantity of data transferred and received by the nodes is minimized, which results in significant power conservation. The nodes are also designed to produce and forward multimedia data, such as a silhouette or foreground image, when requested by system users.

The cluster head node forwards the cluster's data as well as its own data back to a sink server. Data received from different nodes are gathered at the sink and events are extracted based on aggregated data from the nodes. The sink maintains all data in its history database and makes them available to human operators. It is also possible to process these aggregated data to make network-wide inferences and to decide which actions are most appropriate. Furthermore, the sink provides interfaces for end-users and additional applications that require access to the network and events. A standard lightweight messaging protocol capable of transferring both text and image based data is used between the sensor network and the sink server.

3.1.4 Sink Layer

Multimedia sensor nodes that act as cluster hear forward their products as well as cluster’s data back to a server in order to collect all data and produce events according to that data. The connection between layer two and the sink is IP over 802.11 or 802.15.4. The sink keeps all the data in its history database and presents to its users. It is also possible to use this database for further processing like event generation, tracking etc.

Sink acts as an isolation and interaction environment between the users of the system and the underlying sensor network. Apart from conceptual data transfer from the sensor network the sink also manages requests coming from system users. The users are able to request additional data about the threat, like silhouette, picture or even complete view of the scene. Those requests are forwarded to the network via sink server and replies coming back are directed to the requester again by means of sink server.

3.2 System operations

The system performs 2 (two) main pre-operation tasks. First, the system trains itself to learn the features of each object category in order to perform classification. The training data set is composed of images that are cropped from previously recorded videos of the area to be monitored. Secondly, the system produces a background model that is updated statistically. This model is used as a basis for object detection. The background view should contain as few variable elements as possible and the model should also be adaptive, so that new stationary objects can be added to the background view.

The operations performed by the system are illustrated in Fig. 3.2. The first operation is detection of an intruder via scalar sensors, and then the related multimedia sensor is activated. The presented architecture employs PIR, acoustic, and vibration sensors to detect motion, environmental sound, and vibration. These passive sensors are low cost and consume very little power. They provide digital "high" signals to the sensor node. Signals coming from the scalar sensors and the current status of the sensor node are fused, and then the sensor node uses a rule-based decision-making process to activate the camera. Simultaneously, multimedia sensor nodes check if the signal coming from the scalar sensors is persistent, in a predefined period of time. The pseudo codes for the algorithms developed for this proposed system are given in the next section.

Awakened multimedia sensor starts capturing medium resolution images. In order to extract the object from the captured image, background subtraction is used. The current frame is subtracted from the background model and foreground is produced as the result. After that the background model is updated so that the system adapts the changes in the background.

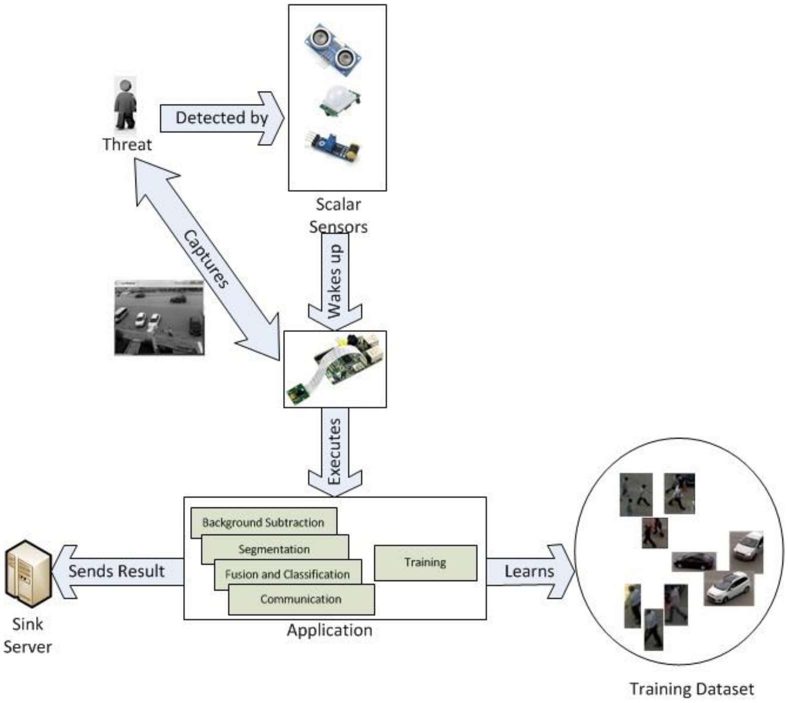


Figure 3.2 System operations

Extracted foreground objects primarily contain defects and noise that cause problems during classification. Such defects are repaired using “opening/closing” morphological operations. Noise is also removed using a median filter. Right after post processing operations, foreground objects become more differentiated and easy to classify. At the end Region of Interest or Minimum Bounding Rectangle that contains clear foreground object is extracted from the scene.

After post processing of a foreground object, its features are extracted. Simple shape-based features of the object, such as the *width/height* ratio, dispersedness which shows complexity of the shape and calculated by *perimeter*^{2}*/area*, and blob area ratio which is the ratio of the total blobs area in MBR to the area of MBR, are calculated. In order to determine the object class based on those features, SVM is applied. SVM is used at this stage because of its good performance and compatibility. The extracted object’s features are matched with features stored in the training data set, and then the node determines how to label the new object. At this stage various classification techniques with multiple feature sets are employed in experiment used to compare their performances; such experiments are described in detail later.

The proposed system also supports shape-based classification with local invariant features. In order to do so SURF is extracted from the training data set and the BoW method is used to store those features during the training phase. The SURF of the new object is extracted according to the results of shape based classification via SVM and is then matched with the bag of SURFs again using SVM. The details of this process are described in the next section. The contribution of invariant features to the proposed system’s classification process is discussed in the experiments section.

3.3 Network

The network is composed of sensor clusters. Within each cluster a sensor node is setup as the cluster head. All other sensor nodes forward their events and data to the cluster head via low-power, low-bandwidth radio devices using IEEE 802.15.4 (ZigBee) standard. Note that ZigBee provides a line of sight up to of 1500 m. at outdoor conditions and 250 Kbps. RF data rate. In order to handle this each cluster is considered as a separate Personal Area Network (PAN) with a separate PAN ID. The cluster head node acts as the coordinator device that provides network synchronization by polling nodes, and the other nodes are configured as end device that rely on the coordinator. A membership is established between the end devices and a coordinator using Pan ID setting.

The modules operate at transparent mode which means that they act as a serial line replacement. The data received from the wired interface of the module (e.g. Universal Asynchronous Receiver/Transmitter-UART) is immediately queued for RF transmission. Besides, the modules also operate at Broadcast mode in order to prevent acknowledgements of packet receptions. In this way the network traffic is decreased and the senor life times are prolonged. The destination addresses in this mode are set as;

DL (Destination Low Address) = 0x0000FFFF
DH (Destination High Address) = 0x00000000

In proposed architecture the ZigBee interfaces are responsible for two types of traffic. The first one is the transfer of extracted conceptual data to the sink station. The structure of this conceptual data is given in Table 3.1. The second traffic type is the requests from the operators to the sensor network. The operators' additional requests for more complex data are transferred to the destination sensor node via ZigBee interfaces. Proposed command structure is given in Table 3.2.

In the network architecture of the proposed system installation of the Global System for Mobile Communications (GSM) network interface in the sensor node is suggested in order to provide the additional capability to forward its data directly to the sink station. Continuous communication via this interface which is operating at 900MHz band is not foreseen, because it requires excessive power consumption (more than 500mA current). On the other hand, this communication capability is planned for use under 2 circumstances. It is firstly considered to be a backup connectivity interface for the sensor node. When the node’s route to the sink via the cluster fails, it can use this interface to forward its text-based events directly to the sink. In addition, the sensor node uses this interface in order to transmit more complex data, such as a foreground image, video frame, or even video stream, whenever demanded from the sink.

Table 3.1 Message Structure and Field Descriptions for Concept Data

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
SensorID																ObjectID															
Timestamp																															
Location-X													Location-Y											Cls		N					

Sensor ID	Unique ID of the sensor that produces the data
Object ID	Unique ID of the detected object
New Object	Field to determine if the object is previously detected
Class	Classification Result for the detected object
Location	The (x,y) coordinates of the object on the video frame
Timestamp	The timestamp of the detection in epoch format

Table 3.2 Message Structure and Field Descriptions for Requests

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">SensorID</td><td colspan="16">ObjectID</td></tr>
<tr><td colspan="32">Timestamp</td></tr>
<tr><td colspan="24">UserID</td><td colspan="4">Type</td><td colspan="4"></td></tr>
</table>

Sensor ID	Unique ID of the destination sensor that the data is requested from
Object ID	Unique ID of the requested object
Type	A numeric value that represents the type of requested data (Siluhette, Picture, Snapshot, Stream…)
Timestamp	The timestamp of the request in epoch format

CHAPTER 4

IMPLEMENTATION

The proposed architecture has been tested using a test bed that was developed for the present study. The test bed includes both hardware and software components. The sensor node, camera sensor, scalar sensors, and sink server for collecting events from the sensor network constitute the hardware component of the implementation.

The software components consist of the following elements:

- C++ application for object detection and classification in the sensor node;
- OpenCV open source library [55] for image processing operations;
- Gloox Extensible Messaging and Presence Protocol XMPP (RFC 6120) client library [56] for messaging;
- wiringPi C++ library [57] for using general purpose I/O (GPIO) ports of the sensor node;
- User space Video4Linux (uv4l) video drivers at the sensor node;
- VideoLan (VLC) at the sensor node for video streaming;
- Openfire XMPP Server[58] for collecting sensor events;
- Spark [59] XMPP client application for operators;
- YAAFE audio features extraction library;
- SOX cross-platform audio conversion utility.

The architecture of the test bed and between-component communication mechanisms is shown in Fig.4.1. Implementation is designed to detect 3 object classes: people, groups, and vehicles. The communication between the cluster head and its nodes is serial data transmission over ZigBee interfaces. On the other hand cluster head can be connected to the sink station using Internet Protocol (IP) over ZigBee, GSM or Wi-Fi interfaces. The sink station is connected to an access network to forwards messages to its users.

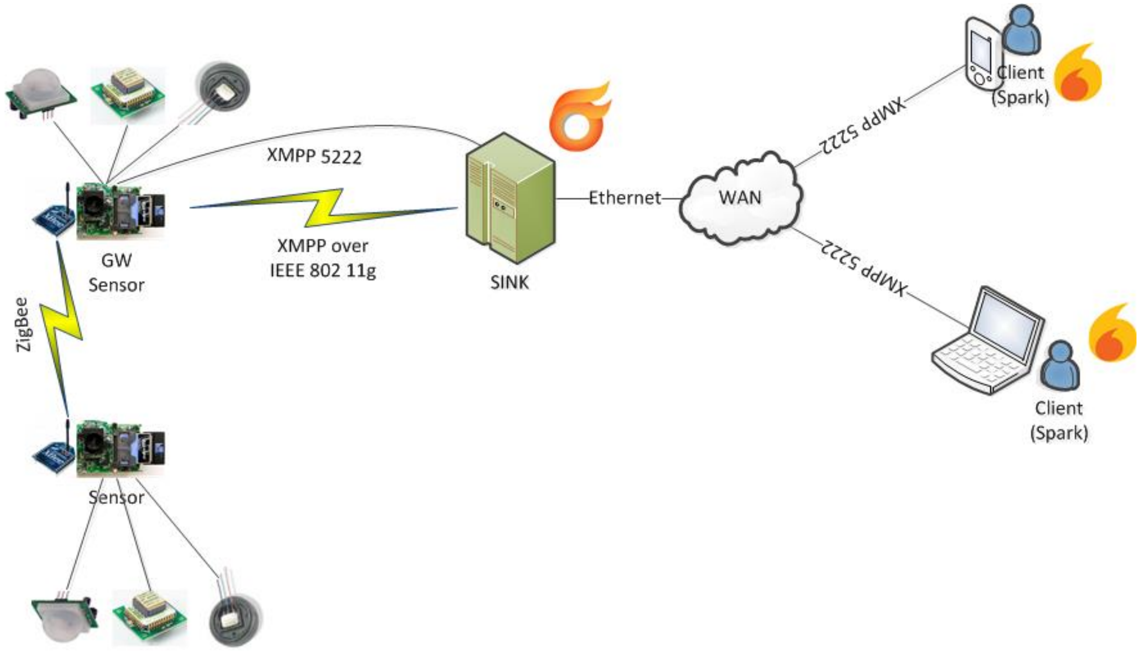


Figure 4.1 Test bed structure

4.1 Sensor Node

Sensor node hardware is based on a Raspberry Pi (RPi) 512-MB Model B board, which is shown in Fig. 4.2, and is setup so that the board includes the following hardware components:

- ARM1176 700MHz processor,
- Graphical processing unit (GPU),
- 512 MB SDRAM shared with GPU,
- SD card slot for on board storage,
- On board 10/100 Mb Ethernet port,
- 2x USB 2.0 ports,
- 1 CSI input connector for the camera module,
- Video and audio outputs,
- GPIO ports,
- 5V 700-mA microUSB power requirement.

The following additional peripherals were installed on the board to fulfill its functions:

- Motion sensor (PIR),
- Acoustic sensor (AS),
- Vibration sensor (VS),
- Raspicam camera module,
- Xbee ZigBee (IEEE 802.15.4) adapter,
- 4400-mAh 5V 1A power bank,
- Microphone,
- Wi-Fi dongle.

An 8-GB SD card is used for internal storage and for installing the Raspbian operating system, a Debian-based free system optimized for the RPi hardware. A Wi-Fi dongle is installed in order to connect the device to the management network. A 5-mega-pixel resolution serial camera module is also connected to the board.

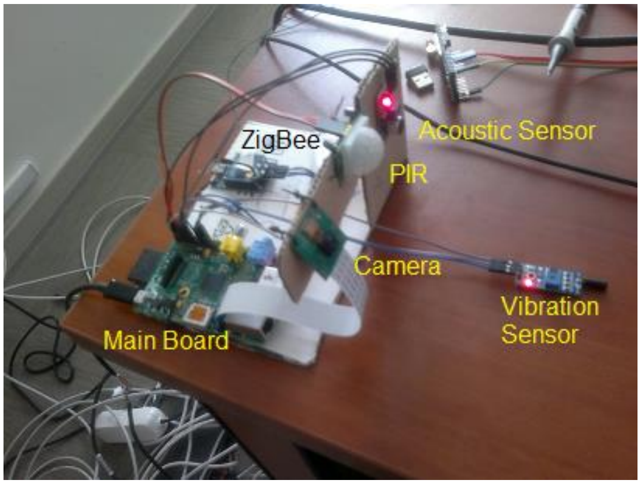


Figure 4.2 Sensor node

The PIR sensor is wired to the device's GPIO port and its output is read. Depending on the brand, the scalar PIR sensor is able to detect motion within a range of 1 to 10 meters. In addition, an AS is connected for detecting environmental sound and VS is connected to detect environmental vibrations. As with the PIR, the other sensors are attached to

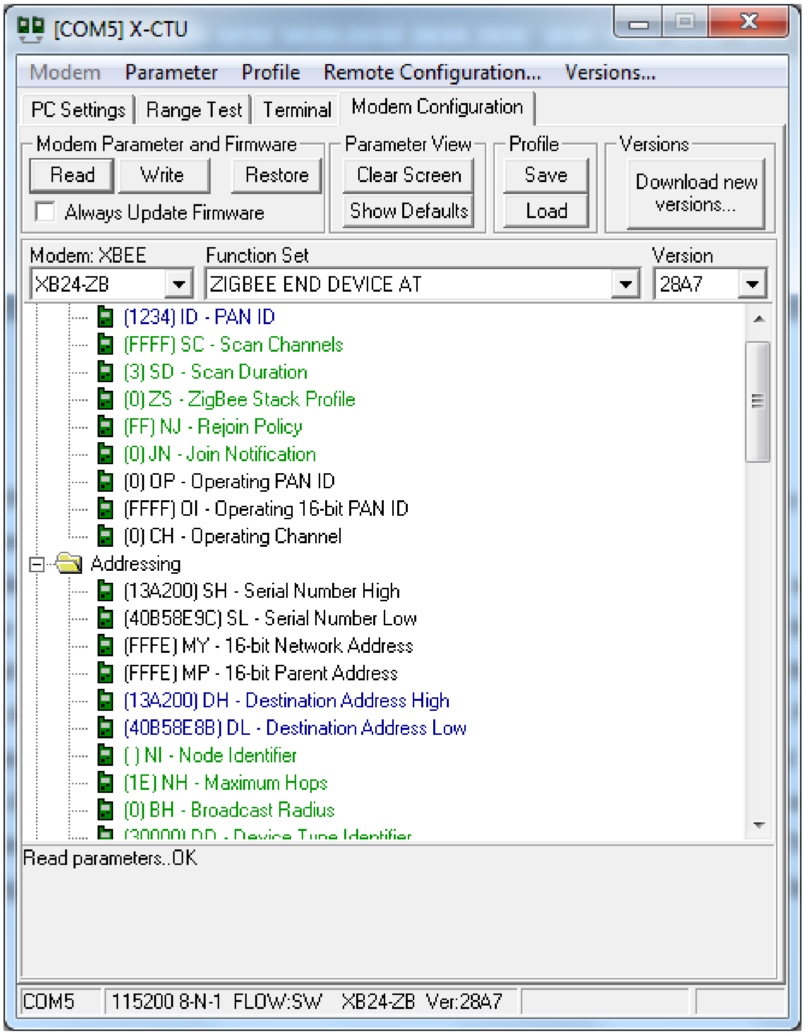
GPIO pins. Before turning the camera on and starting object detection, the application waits for a persistent “high” signal from the scalar sensors. The microphone is connected to the SPI (Serial Peripheral Interface) of the GPIO. The details related with microphone and audio will be given in audio extension section of this chapter.

Sensor nodes are equipped with ZigBee modules. According to the proposed system’s architecture, the sensor nodes are connected to a gateway node configured as a cluster head. This cluster head forwards all cluster events and data to the sink. Sensor to sensor communication, as well as gateway to sink communication, is completed via ZigBee interfaces. The Xbee modules on the sensor nodes are configured via “X-CTU” software provided by the vendor of the modules. Sample configuration is given in Fig. 4.3. The sensors send their data to the gateway node using broadcast serial messages via ZigBee interfaces; however, due to the messaging protocol, communication between the gateway node and the sink server is setup as IP over point to point protocol (PPP). In order to implement this communication “ppp” daemon of the linux is configured and used. After execution of the daemon, a ppp network interface is created within the sensor node as shown in Fig.4.4.

In the test bed sensor nodes GSM interfaces are simulated using Wi-Fi dongles. When additional information, such as a foreground object or video frame, is requested from a specific sensor node, the node forwarded this information directly to the sink using its Wi-Fi interface. Those Wi-Fi interfaces also help us in device management operations.

4.2 Application

A C++ application is developed to implement object extraction and classification operations at the sensor node. The application uses the OpenCV library for image processing functions. It also uses the Gloox XMPP Client library to send messages to the sink and receive commands from users. An additional library called “wiringPi” for reading scalar sensor output is also included.



Modem: XBEE XB24-ZB

Function Set ZIGBEE END DEVICE AT

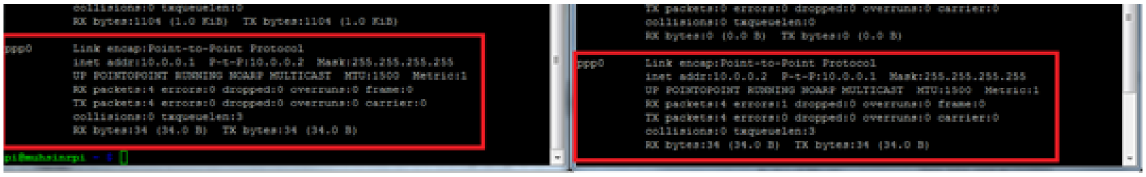
Version 28A7

```
(1234) ID - PAN ID
(FFFF) SC - Scan Channels
(3) SD - Scan Duration
(0) ZS - ZigBee Stack Profile
(FF) NJ - Rejoin Policy
(0) JN - Join Notification
(0) OP - Operating PAN ID
(FFFF) OI - Operating 16-bit PAN ID
(0) CH - Operating Channel
Addressing
  (13A200) SH - Serial Number High
  (40B58E9C) SL - Serial Number Low
  (FFFE) MY - 16-bit Network Address
  (FFFE) MP - 16-bit Parent Address
  (13A200) DH - Destination Address High
  (40B58E8B) DL - Destination Address Low
  ( ) NI - Node Identifier
  (1E) NH - Maximum Hops
  (0) BH - Broadcast Radius
  (300000) DD - Device Type Identifier
```

```
Read parameters..OK
```

```
COM5   115200 8-N-1  FLOW:SW   XB24-ZB  Ver:28A7
```

Figure 4.3 Xbee configuration via X-CTU software



```
          collisions:0 txqueuelen:0
          RX bytes:1104 (1.0 KiB)  TX bytes:1104 (1.0 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.0.0.1  P-t-P:10.0.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:34 (34.0 B)  TX bytes:34 (34.0 B)

pi@muhsinrpi ~ $
```

```
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.0.0.2  P-t-P:10.0.0.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:1 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:34 (34.0 B)  TX bytes:34 (34.0 B)
```

Figure 4.4 PPP interfaces over Xbee

4.2.1 XMPP and Related Components

In order to enable messaging between the components of the architecture, XMPP protocol is preferred. XMPP is a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data. The advantages of using XMPP in our architecture are [59];

- Open: Free, easy to implement, lots of implementations are available.
- Standard: Documented in RFCs 3920, 3921, 6120, 6121 and 6122.
- Decentralized: Anyone can tailor and run their own XMPP server that fits his application requirements.
- Extensible: It is possible to implement extensions on top of the core protocols.
- Flexible: Its XML nature makes it available for a wide range of applications.

The XMPP components used in our architecture are;

Openfire: Openfire is an open source (Java Based) cross-platform real-time server based on the XMPP. It is available for different type of operating systems and licensed under the Open Source Apache License. A number of java plugins are available for providing additional capabilities to Openfire. It is also possible to implement and deploy custom plugins with specific functions related with the application. Openfire is chosen as the server process running at the sink. It will collect XMPP messages coming from the sensor network and operators.

Spark: Spark is also an open source (Java Based), cross-platform messaging client which uses XMPP protocol. Spark is supported by the same community of Openfire as well. It is chosen for operators to communicate with the sink.

Gloox: Gloox is an open source XMPP client library, written in clean ANSI C++. Since our sensor node application is written in C++, the messaging modules are injected in the node’s application using Gloox libraries. Gloox provides enough set of functionalities in order to fulfill the messaging requirements of the sensor node

Before mentioning the details of the application, it would be useful to talk about two important XMPP extensions, Multi User Chat (MUC) and File Transfer. XEP-0045 defines MUC as an XMPP extension. Multiple XMPP users can exchange messages in the context of a room or channel, similar to Internet Relay Chat (IRC). In our architecture the gateway sensor node and operators are joined to a predefined room by using accounts that are created on Openfire server for them. Gateway sensor node sends its own events as well as its sensors’ events to that room. Operators who are authorized to connect that room will be able to follow those events. Sample XMPP Stanzas of MUC are given in Table 4.1.

The File Transfer is another extension which is specified in XEP-0096. This specification defines a profile of the XMPP stream initiation extension for transferring files between two entities. File transfer is used to provide additional data to operators whenever requested. Sensor application is designed to receive those requests and prepare the requested data as image file. The sensor node sends this file to the requestor’s client application by using XMPP stanzas. During the file transfer, the Openfire server acts as proxy and all transfers are performed through the server. Point to point file transfer between the operators’ client application and the sensor node is not allowed.

Following settings are configured on the sink server.

- Operator and sensor nodes accounts are created.
- One chat room is created for accounts to send and receive messages.
- TCP ports 5222 and 7777 are configured for messaging and File transfer respectively.
- Logging is fully enabled.
- DNS records are configured so that the clients are able to reach the server and its

MUC service

- History transfer for MUC is disabled to prevent excessive traffic.

4.2.2 Object Identification Application

The application on the sensor node first reads its settings from a configuration file. Those settings are given in Table 4.2. The node uses those settings to connect itself to the XMPP server and join the chat room. Prior to surveillance, the application runs a training process. The training images are stored on the sensor node’s file system. As the system was designed to detect people, groups, and vehicles, sample images of those categories, as given in Fig.4.5 were processed during training. The training image dataset was generated using sensor node video recordings. During the training phase shape-based features in those images were calculated, including the width/height ratio of an object’s bounding box, compactness of the bounding box, and the ratio of the total blob area to the bounding box area.

Table 4.1 MUC XMPP Stanza

XMPP Stanza	Description

<message
from='gw1@sensornetwork'
id='gw1'
to='events@conference.sensornetwork'
type='groupchat'>
<body>Sid:11,Person.,New,10.10.16 14:12</body>
</message>
	MUC message from “gateway1” sensor node to all occupants of the “events” chat room. The original message belongs to sensor node with id 11.

<message
from="events@conference.sensornetwork/gw1'
id='E36F45B8-DE06-4534-94AD-C5ED294E'
to='operator1@ sensornetwork   type='groupchat'>
<body>Sid:11,Person.,New,10.10.16 14:12</body>
</message>
<message
from='events@conference.sensornetwork/gw1'
id='ACA9201-2BA0-4A20-98D4-B9CB8582'
to='gw2@conference.sensornetwork'
type='groupchat'>
<body>Sid:11,Person.,New,10.10.16 14:12.</body>
</message>
	Groupchat message in the previous line of this table is distributed to all occupants by the MUC service.

Table 4.2 Node Settings

Setting	Sample Value	Description
id	Gw1	Unique id of the sensor node
xmpp_server	10.0.0.1	IP Address of the XMPP server
xmpp_domain	Sensorevents	Fully qualified domain name of the XMPP Server
xmpp_user	Gw1	XMPP Username to login the server
xmpp_passwd	password	Password for login
xmpp_room	events	MUC room to join and forward messages

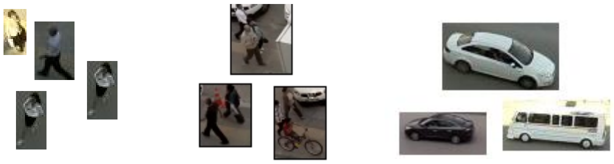


Figure 4.5 Sample training images for people, group and vehicle classes

The application is designed to support shape-based features as well as local invariant features, so as to maximize classification accuracy. SURF was chosen for this application because of its efficiency. The SURFs of the training dataset images are extracted during the training phase and their histogram is prepared using BoW. Those bagged SURFs are used to perform a second level classification via SVM. In this way the cascade classification mechanism is implemented in the sensor node.

After training, each sensor node creates a background model of its field of view. Constructing the model, the node checks its scalar sensors and attempts to decide if the camera should be opened and object detection should begin. A continuous “high” signal from any of the scalar sensors indicates that there exists a moving object. Immediately after the sensor node turns the camera on and captures frames, it performs image processing operations on the captured frames. BS, morphological operations, and image segmentation are applied to extract the foreground object. If a persistent scalar signal doesn’t exist, the application turns the camera off and enters sleep mode. In this way the node does not capture and process unnecessary frames and saves energy. During this idle period the sensor node updates its background model in order to adapt to any changes in its field of view. The overall algorithm is given in Algorithm 1 and the decision-making process is outlined in Algorithm 2.

Algorithm 1: Main application block

```
INPUT: PIR, AS,VIB sensor outputs
OUTPUT:Detected object report

if (cameraStatus(PIR,AS, VIB,Status)==OFF)
 //Deciding on camera status.
  Sleep()
else
    openCamera()
    capture()
    update_background_model()   //keep bg. up to date
    extract_foreground_objects()
    for each foreground_object
      if (object_area>threshold)  //filter small objects
         getClass(wh_ratio,comp,blob_ratio,obj_img)
        sendResult() //send results to sink or GW sensor
        if (object_already_detected)   //detected before ?
          update_object_attributes()
        else
          add_active_object_list() //new object!!!
        endif
      endif
   endfor
endif
```

After detection of a foreground object, its features are extracted. As mentioned earlier, 3 shape-based features are calculated and classification is performed using those calculated values. In the end, the best matching label from the training set is extracted. According to our experiments, vehicle type objects are well categorized; however, confusions occur at categorization of group and people type objects.

Second-level classification is done when first-level classification results do not indicate the vehicle type. At this stage SURF descriptors of the detected object are extracted and matched with bags of SURF descriptors of the training data set using SVM. Object class is determined according to SVM results. The classification process is roughly described in Algorithm 3.

Algorithm 2: Deciding Camera Status

```
INPUT:Camera state, PIR, AS, VIB
OUTPUT: New status

cameraStatus (PIR,AS, VIB, Status)
 if(PIR or AS or VIB) //High signal from scalar sensors
   if(Camera is OFF)
     wait()//No rush. Hold on
     if(PIR or AS or VIB) //Still active
       return ON //Turn camera on
     else  //wrong alarm
       return OFF //Keep camera off
     endif
   else if(Camera is ON)
     return ON      //Continue detection by keeping camera on
   endif
  elseif (!PIR and !AS and !VIB)
    return OFF //No alarm, turn camera off.
 Endif
```

Algorithm 3: Classification

```
INPUT: The width/height ratio of object, the ratio of the area of the
blob to the area of MBR, and compactness of the MBR.
OUTPUT:  Object class

getClass(wh_ratio,compactness,blob_ratio,obj_img)
label=svm_predict(wh_ratio,compactness,blob_ratio) //detect label
  if(label == "vehicle")/ /result is vehicle.
     return(label)
  else  //perform SVM with BoSURF
  return(getSVMBoW(obj_img))
// return label of best matching neighbor
```

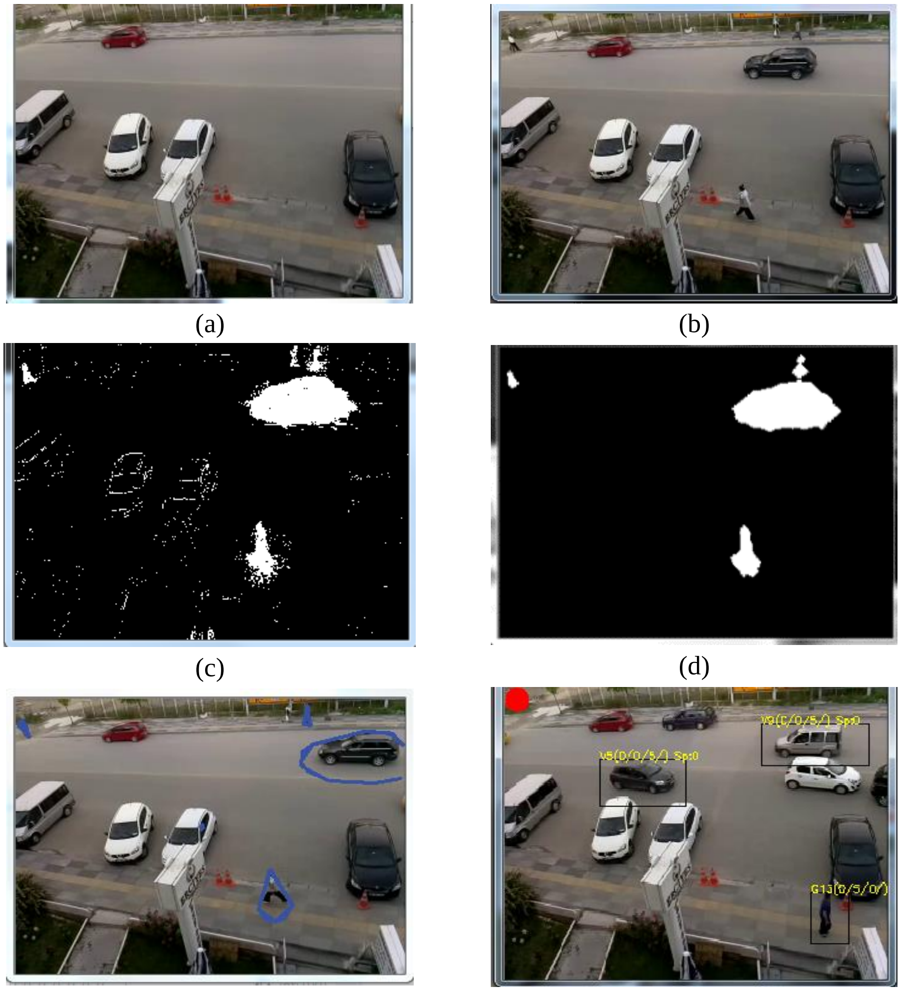
Although it is very common to use local invariant features for object detection, the primary disadvantage of local features is that extraction and matching operations are time consuming. As such, it can be concluded that local features-based supervised classification is not feasible for real-time resource-constrained surveillance systems, i.e. WMSNs. This argument is supported by the present study’s experiments, as described in the next section. For resource-constrained systems we propose limited use of local features during classification. We first attempt to classify objects using SVM and shape-based features, which are more efficient. According to the results of this classification, we utilize SURF with SVM.

Another important reason for choosing such a cascade classification approach is related with sensor node video quality. The proposed system's sensor node's camera is not high resolution; it records low-quality videos of 320 x 240 pixels at 5fps. Extracting sufficient local features from such low-quality video for object classification is a challenging task. The present experiments prove that classification accuracy using only local features is much lower than classification using shape-based features. Overall execution of the application is illustrated in Fig.4.6. The performance of object detection and classification is evaluated by setting different values to the parameters of related OpenCV functions. In this way the most appropriate parameter set is determined. Those parameters and settings are given in Table 4.3.

4.2.3 Classification Methods

In order to determine the object features and the classification method to be used for object identification, a few different approaches are implemented and tested in the application. The details of the tests results are given in the next chapter. Herein it is given information about the implementation of those approaches.

We firstly use k-NN based classification with shape based features. The application initially performs k-NN based training using shape based features of the training images. Moving object's shape based features are extracted and sent to the classification function in which number of nearest neighbors parameter is set to 5. The class of the object is determined as the class of the majority of the returned 5 neighbors.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.6 Object detection and classification steps using the proposed system's application a) Constructing background model, b) Capturing frame, c) Background subtraction, d) Post processing, e) Object extraction, f) Classification

Table 4.3 Parameters of the Application

Parameter	**Description**	**Value**
BS_HISTORY	Length of frame history used for BS	25
NMIX	Number of mixtures for MoG	3
RATIO	Threshold for adding the object to background model	0.89
SVM Classifier	Type of SVM Classifier	N class classifier
Kernel Type	Type of kernel function of the classifier	RBF
# of iterations	# of iterations for median filtering	3

We extend the k-NN based classification function by adding robust features to the process. We select SURF because of its better performance with respect to SIFT, as it is stated in chapter 2. The training function of the application is modified so that the SURF of the training images are extracted and bagged using BoW approach. Those bagged features are then sent to SVM to finalize SURF based training process. Rather than immediately deciding the category of the object according to the k-NN result, we check the number of nearest neighbors of different classes returned at the end of k-NN matching. If 4 or 5 of the nearest 5 neighbors belong to the same category then the classification is stopped and that category is returned. Otherwise a second level classification based of bagged SURFs is performed. In order to realize this second level classification, we sent the RoI of the frame which includes the object to be classified to the function. We extract the SURF of this RoI and perform SVM based prediction by using the bagged SURF of the training images. This second level classification determines the category of the object.

We implement a third classification function which uses SVM and shape based features. During the training process extracted shape based features of the training images are sent to SVM rather than k-NN. This trained SVM is used in the classification function to predict the category of the extracted object. The only difference of this implementation and the first one is the usage of the SVM instead of k-NN.

As fourth approach, we apply the similar extension we made for k-NN based classification function to SVM based classification function. We use bagged SURFs of the training images for second level classification. Since SVM prediction method in OpenCV does not give the categories of the training objects that determine the result, it was not possible to apply the same decision rule we use during k-NN based classification. For that purpose we implement a different rule to initiate the second level classification. We benefit from the results of the previous classification functions in order to build that rule. According to the test results vehicle category is the one with best accuracy. So we assume that if the result of the SVM classification is vehicle category then the classification function is finalized and the category is returned. Otherwise SURF based classification is performed in the same way as explained before and the result of this function is returned.

As the last method we solely perform bagged SURF based classification. The RoI of the object is directly sent to the classification function without extracting shape based features. SVM based prediction is performed and the category of the object is determined.

Another implementation detail is related with the training phase. As it is stated, SVM and k-NN methods are used to train the system using shape based features of training images. We also perform training using normalized shape based features. In order to do that we calculate the mean and standard deviation values for each feature of each category. The normalized values of each feature are calculated as shown below.

$$\mu = \frac{\sum_{i=1}^{k} f_i}{k}$$

$$\sigma = \sqrt{\sum_{i=1}^{k} (f_i - \mu)^2} \tag{4.1}$$

$$f_n = \frac{f - \mu}{\sigma}$$

where μ is the mean value of the features $\{f_1, f_2, \ldots f_k\}$, k is number of objects of certain category, σ is the standard deviation and f_n is the normalized value of feature f. Instead of extracted features, normalized feature values are used during training. In this way we aim to see the effect of normalization on the performance of the classification.

During implementation of all functions we measure the delay by calculating the number of clock ticks between start and end of the method. This gives us opportunity to compare the efficiency of the implemented methods with respect to the processing time. Since the host platform is a resource constraint one, the selected method should be efficient enough to be executed on such platform.

4.2.4 Run-time Storage

In order to store the detected objects' attributes, we implement a linked list data structure. The list is composed of object structs which hold attributes given in Table 4.4.

Table 4.4 Record Structure that Holds the Attributes of Detected Object.

Attribute	Type	Description
ID	int	Unique ID produced for that object.
category	smallint	Category of the object
whRatio	float	width/height ratio of the bounding rectangle of the object.
compactness	float	Compactness value of the bounding rectangle.
areaRatio	float	The ratio of the sum of blob's area of the object to the area of the bounding rectangle.
location	Point	x,y coordinates (in pixels) of the upper left corner of the bounding rectangle of the object.
lastDetectionTime	Timestamp	The last time value when the object is seen.
speed	smallint	To be used for detected speed of the moving object.
snapshot	Matrix	Image of the object extracted from the frame.
next	pointer	Pointer showing the next node in the list.

When a moving object is and its category is determined, the linked list is checked if the object is already inserted into the list. To perform this check we use the category of the object, last detection time and the position of the object in the frame. If a recently detected object exists in the linked list with the same category of the detected object and close to the detected object's location, it is considered that the detected object is previously inserted in to the list. In this case the attributes of the object is updated. Otherwise a new node is created for the detected object and that node is appended to the end of the list.

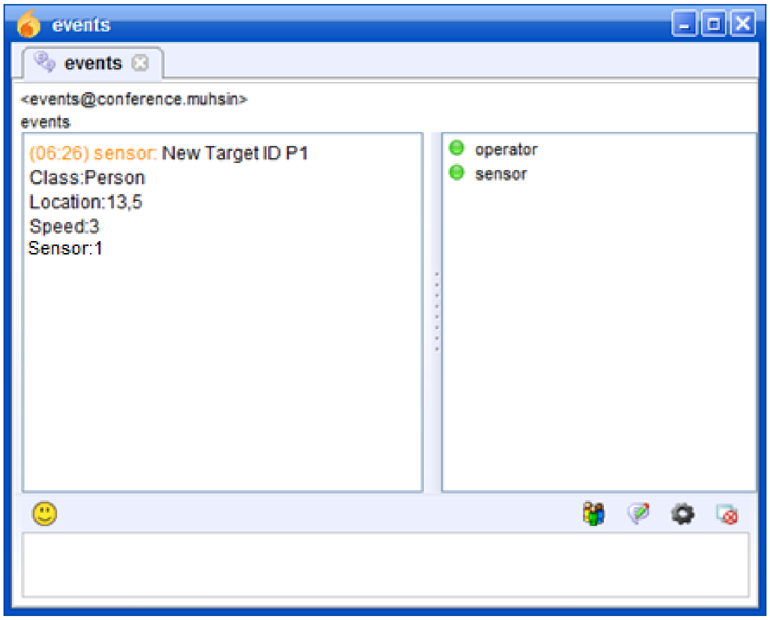
In order to control continuous grow of the linked list, a periodic check is performed by a separate thread. This thread checks the last detection timestamp values of each object in the list, calculate the age of the object and compare it with a predefined threshold. If the age of the node is greater than the threshold the node is deleted from the list. In this way unnecessary storage of passing objects is prevented.

4.3 Sink Station and Messaging

Once the application on the sensor node makes a decision concerning a threat, it forwards this information to a server system; for that purpose sensor nodes use their cluster heads as a gateway. The gateway node collects messages from its cluster and forwards them to the sink station. At the physical layer the sensor node is connected to the sink via a ZigBee interface. PPP is configured as the data link protocol.

The sink is connected to a wide area network (WAN) using its network adapter, which can be a wireless adapter, network interface card or a modem device. The client applications, which are connected to the same WAN, are authorized to connect to the sink. Direct communication between the clients and the sensor network is not prohibited. In order to enable messaging between the cluster head, sink, and client application, XMPP is preferred.

To enable XMPP communication the Gloox C++ XMPP client library is installed on the sensor node. At the sink station, the Openfire XMPP server is installed and a chat room is configured to join the gateway sensor nodes and the human operators. The sensor node connects to the XMPP server using its jabber account over the 5222 TCP port. After successfully connecting, the node joins that chat room. Whenever the sensor node detects and analyzes a threat, it encapsulates the results in MUC messages and sends them to the chat room as shown in Fig.4.7. The operators connect to the sink server and join the chat room via the Spark client application. Once operators join to the chat room, they are able to follow the real-time messages coming from the sensor network. Messages contain data related to a detected object, including class, location, and speed.



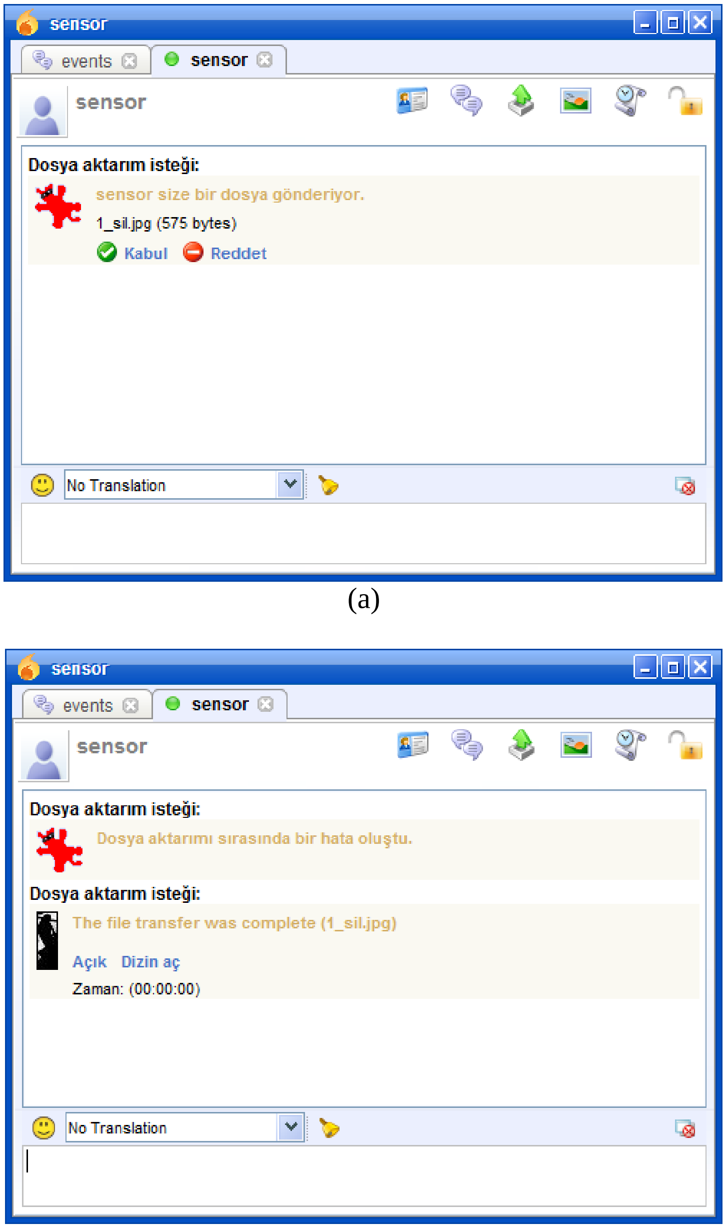
<events@conference.muhsin>
events

(06:26) sensor: New Target ID P1
Class:Person
Location:13,5
Speed:3
Sensor:1

Figure 4.7 MUC room

According to the proposed architecture, operators are able to request additional data from the sensor network. Using the MUC window, an operator is able to request a silhouette or foreground image from the sensor node. To do so, an operator sends a command to a specific sensor with the sensor ID and object ID parameters. For example, when an operator requests the silhouette of object with ID: “1” from a sensor with ID: “sensorgw”, he/she sends the message, “sensorgw,sil,1”, to the chat room, or, if an operator wants the foreground image he/she sends the “sensorgw,pic,1” command.

The command is received by the MUC service of the sink server and the server forwards this message to all other occupants of the chat room including the gateway sensor nodes. The gateway sensor node parses the message, checks the destination sensor id, and forwards it. According to the type of data requested the destination sensor node prepares a file in JPG format that contains either a black & white or color image of the object. When the file is ready the sensor node uses its alternative GSM interface (which was simulated using a Wi-Fi interface in our test bed) to send the file back to the requesting client application; sensor nodes use their XMPP accounts for that purpose. The file transfer proxy service of the Openfire server handles the file transfer operation, so that direct communication between the client application and the sensor network is prevented. As soon as an operator accepts the file transfer request, the sensor node begins uploading the prepared file in XMPP stanzas, as shown in Fig. 4.8.



Dosya aktarım isteği:

sensor size bir dosya gönderiyor.

1_sil.jpg (575 bytes)

Kabul Reddet

(a)

Dosya aktarım isteği:

Dosya aktarımı sırasında bir hata oluştu.

Dosya aktarım isteği:

The file transfer was complete (1_sil.jpg)

Açık Dizin aç

Zaman: (00:00:00)

(b)

Figure 4.8 File transfer a)Transfer request is received from the sensor, b)Transfer completed.

An advantage of using XMPP is makes it possible to provide a standard interface to other applications that need to process the events in the network. Using an open protocol, new applications that are developed for future requirements are able to fetch the events in real- time or browse the event history. In this way, the system provides event notification services not only to system operators, but also to programmers.

4.4 Sensor to Sensor Communication

As stated above, sensors are connected to the gateway sensor node via ZigBee interfaces. Within the application, a serial messaging thread is implemented in order to forward the events to the gateway sensor. The gateway collects all the events coming from its cluster via broadcast messages, encapsulates those events in XMPP stanzas, and forwards them to the sink server. The same mechanism is also used for command transfer. When the operator sends a command to a sensor using the client tool, the command is forwarded to the gateway sensors via MUC messages. The gateway sensor forwards the command to the cluster using serial broadcast messages that include the target sensor’s id. The commands are processed by the destination sensors, and necessary output files are prepared and sent, as previously explained.

4.5 Audio Extension

The sensor node is equipped with an analog microphone device which is connected by using the SPI pins on GPIO of the RPi. In order to read audio data through the microphone an analog to digital convertor (ADC) integrated circuit is used. The integrated circuit we prefer for A/D conversion is MCP3008 which is 10 bit 8 channel convertor. We choose 8 channel integrated circuit because of its availability when connecting additional scalar devices. The connection between RPi GPIO, MCP3008 and the microphone is given in Table 4.5. The sensor node with microphone is shown in Fig.4.9.

The microphone is added as second multimedia device. In this way it will be possible to collect the environmental sounds and perform additional classification based on this collected audio data. 3 types of audio data is to be categorized; human, animal and vehicle. This audio classification is used for 2 purposes. First; audio classification result is used to activate the camera. If the audio categorization result is human or vehicle then the camera is activated. The details related with this mechanism are given in latter paragraphs. Second; audio classification result is fused with the video processing result in order to increase the overall performance of the node's classification. The effect of audio classification on the overall performance is shown in the next chapter.

Table 4.5 Raspberry, MCP3008 and Microphone connections

MCP3008	RPi GPIO	Microphone
VDD (16)	5V (2)	Vcc
VREF (15)	5V (2)	
A Ground (14)	Ground (6)	Ground
Clock (13)	SPI Clock (23)	
Data Out (12)	SPI MSIO (21)	
Data In (11)	SPI MOSI (19)	
Chip Select (10)	SPI CE0 (24)	
D Ground	Ground (6)	
Channel 0 (1)		Output

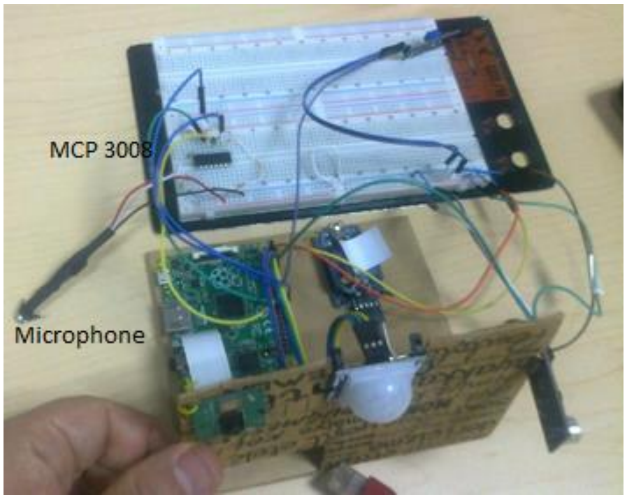


Figure 4.9 Sensor node with microphone

4.5.1 Audio Detection Application

An additional application which captures the audio data from the SPI of the RPi and classifies it is implemented on the sensor node. Like main application, this application starts with training as well. For the training dataset we record various sounds of human, animal and vehicle categories by using the node itself. The 13 Mel-frequency cepstral coefficients (MFCCs) of the recorded sounds are extracted and stored in a comma separated value (CSV) file. A CSV file for each category of audio (human, animal and vehicle) is created. The application trains itself by using those CSV files and learns the MFCC features of the categories. SVM is used as the machine learning algorithm in this application.

The application listens the output of the acoustic sensor before capturing the audio data. Before the microphone integration, the acoustic sensor was used to trigger the camera in the main application. After the implementation of the audio application, the acoustic sensor is no more used in the main application. In the audio application its high signal is expected to initiate audio data capture and classification. As soon as a high signal is received from acoustic sensor, the application starts capturing raw audio data from the SPI of the RPi where MCP800 ADC and the microphone are connected.

The application continues capturing audio data for a certain period (2 seconds in the application). According to the frequency of the ADC, I/O speed of the node and its processing power as well, it is buffered raw audio data at nearly 10 KHz frequency (which is approximately equal to low quality Pulse Code Modulation-PCM sound) and 16 bits resolution at the end of the collection period. The MFCC features of the collected data are extracted and sent to SVM to predict its category. Rather than giving the full matrix of MFCC features, mean value of each column in matrix is calculated and vector with 13 scalar MFCC values are used for classification. The result of the classification is then sent to the main application by means of a UDP socket which is also listened by the main application. The flowchart of the audio application is given in Fig. 4.10.

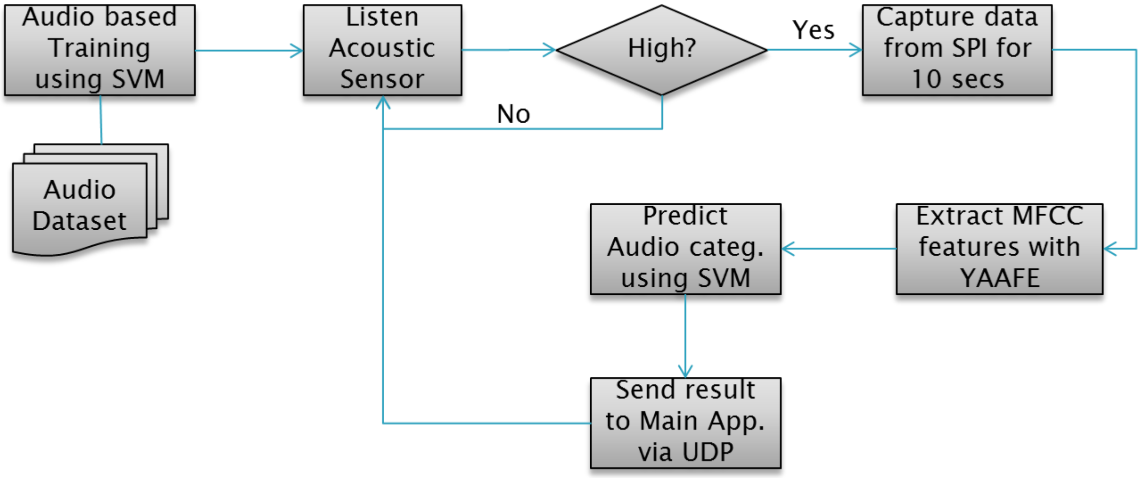

Audio based Training using SVM
Audio Dataset
Listen Acoustic Sensor
High?
Yes
No
Capture data from SPI for 10 secs
Extract MFCC features with YAAFE
Predict Audio categ. using SVM
Send result to Main App. via UDP

Figure 4.10 Flowchart of audio application on the node

4.5.2 Audio Related Modifications in Main Application

According to the audio extension in the sensor node, the main recognition application on the node is modified. The signal coming from acoustic sensor is no longer used to activate the camera. Instead, the categorization result coming from audio application is used as additional criteria to activate the camera. For that purpose, a thread is constructed to receive the result of audio categorization. This thread continuously listens the UDP socket on a specific port. The audio application is expected to write its results to the same port. High level flow chart of the modifications is shown in Fig. 4.11.

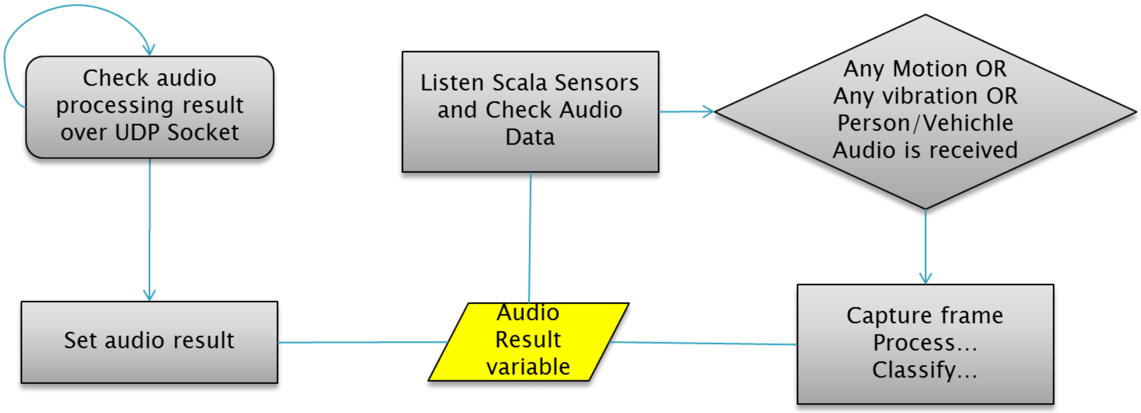

Check audio processing result over UDP Socket
Listen Scala Sensors and Check Audio Data
Any Motion OR Any vibration OR Person/Vehichle Audio is received
Set audio result
Audio Result variable
Capture frame Process... Classify...

Figure 4.11 Modified main application

CHAPTER 5

EXPERIMENTS

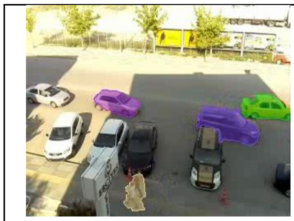
We have tested our architecture with two (2) experiment sets. The first experiment set is related to the proof of the concept and the object detection and classification accuracy. The application has been implemented using numerous classification methods with various features in order to determine their effect on classification accuracy. In this set of experiment we also perform an additional test to see the effect of audio categorization to overall classification performance.

The second experiment set is related to sensor node power consumption. While the sensor node is run in various modes, power consumption is measured in order to determine the performance regarding to power saving of our approach at the sensor node. Furthermore, we measure the delays at each stage of the application to see the most time consuming operations. We also measure the false alarm rates due to the environmental factors. Those false alarms trigger the camera and cause energy waste of battery power.

5.1 Classification Accuracy and Performance

The object classification performance of the proposed system has also been tested by executing the object detection and classification application using street surveillance video recorded using the sensor node. All video files were recorded in H264 format, at 5fps rate and 320 x 240 resolutions. The training data set is composed of cropped images obtained from 4 video files. It consists of images of 285 people, 254 groups, and 270 vehicles. Another video file consisting of 1105 frames (221 secs.) is used for testing. The ground truth of this test video was extracted in XML format using the Sensarea video object editing tool [60], as shown in Fig.5.1.

In all, 5 different models are used during classification as summarized in Table 5.1. In addition to the performance of each classification model, the contribution of the selected features to classification was also analyzed. For the first classification model KNN and shape-based features are selected. The second model also utilizes KNN and shape-based features with the addition of SVM and bag of SURF, so as to determine the contribution of local features to classification. The third model uses shape-based features and SVM. SVM and shape-based features are augmented by bag of SURF for the fourth method. Lastly, to determine the accuracy of using local invariant features alone during classification, the fifth classification model employs SVM and bag of SURF. The same training data set and video file were used to test each method.



```
<frame number="212">
    <object width="53" height="29" type="3" toplefty="83" topleftx="10"/>
    <object width="61" height="28" type="3" toplefty="79" topleftx="181"/>
    <object width="22" height="37" type="2" toplefty="189" topleftx="166"/>
    <object width="13" height="20" type="3" toplefty="218" topleftx="164"/>
</frame>
```

Figure 5.1 Annotated video

The application was executed in a resource-rich PC environment and the time for each method to process the video file was calculated, which provide data concerning real-time usage of each method in the sensor node. We calculate precision and recall rates for each object type using the ground truth. The classification experiment results are shown in Table 5.2. In addition, for each method, the precision/recall curves for each object category is produced and presented in Fig.5.2. We choose overlapping ratio between minimum bounding rectangles (MBR) of objects in the ground truth data and the MBRs that we extracted in video frames as the threshold of the detection.

Table 5.1 Features and Classification Methods

Model	Features	Method
1	Width/height ratio, Compactness, Blob ratio	KNN
2	Width/height ratio, Compactness, Blob ratio, Bag of SURF	KNN SVM
3	Width/height ratio, Compactness, Blob ratio	SVM
4	Width/height ratio, Compactness, Blob ratio Bag of SURF	SVM SVM
5	Bag of SURF	SVM

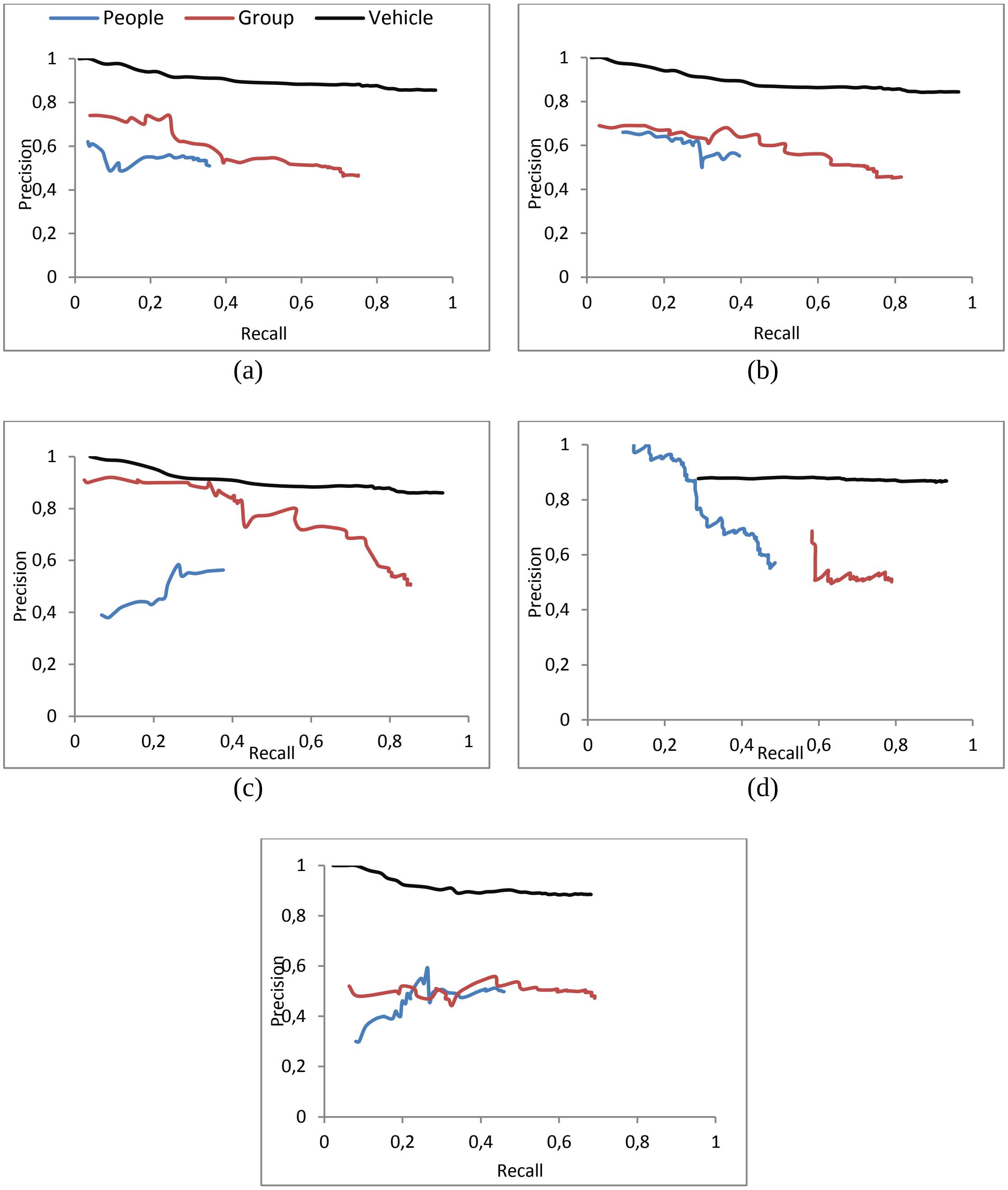
The results of the experiment set show that shape-based features are satisfying. Although local features facilitate classification of people, according to the results of models 2 and 4, their use provides two (2) primary disadvantages that are observed with model 5. Firstly, using local features alone for classification yields poor performance with low-resolution videos. The number of descriptors that are extracted from low-quality images is not sufficient for a reliable classification. Secondly, extraction of local features and matching the features are time consuming. As the application is planned to be executed in real time on a resource-constrained platform, the classification model needs be compatible with those constraints.

In total, two (2) factors are observed to affect the classification performance of people. Firstly, shadows increase the false positive rate and decrease the precision and recall rates of the people type objects; therefore, we think shadow removal techniques must be applied before beginning the classification process. Secondly, the application confuses groups of people and individuals. Some individuals (according to ground truth) were classified as groups of people and vice versa. Changing the training data set’s size and normalizing shape-based features during training had a negligible effect on overall classification performance. We conclude that it is necessary to use additional shape-based features to differentiate individuals form group of people.

Table 5.2 Classification Performance and Accuracy

Model	**Duration**	**Performance**			
		Object	**Recall**	**Precision**	**F-Score**
1	56 s	Person	27%	40%	0,32
		Group	45%	25%	0,32
		Vehicle	90%	92%	0,90
		Average	54%	52%	0,52
2	59 s	Person	30%	39%	0,33
		Group	44%	32%	0,37
		Vehicle	96%	91%	0,93
		Average	56%	0,41	0,47
3	56 s	Person	20%	58%	0,30
		Group	65%	30%	0,41
		Vehicle	93%	87%	0,89
		Average	59%	58%	0,58
4	**62 s**	**People**	**39%**	**44%**	**0,41**
		Groups	**43%**	**48%**	**0,45**
		Vehicles	**92%**	**93%**	**0,92**
		Average	**58%**	**61%**	**0,60**
5	96 s	People	35%	18%	0,23
		Groups	42%	22%	0,28
		Vehicles	50%	94%	0,65
		Average	42%	44%	0,42

In order to improve the performance of classification according to the conclusions we face in the previous tests, we decide to unify person and group of people categories. In this case the application classifies two categories of objects, person/group and vehicle. When application categorizes an object as either person or group, and the same object is annotated as person or group in the GT then we consider it as a true positive based on its bounding box on the frame. We perform test by using model 4 which is evaluated as the most successful method at the end of previous tests. The results show that the recall and precision rate of the unified category is %58 and 77% which is much better than the rate of both of the categories in the previous tests.



(a) (b) (c) (d) (e)

Figure 5.2 Precision vs. Recall curves of a)Model-1, b)Model-2, c)Model-3, d)Model-4, e)Model5

5.2 Accuracy of triggering camera

In order to measure the reliability of the scalar sensors used in our sensor node, we have performed a number of experiments on the sensor node. The purpose of these experiments is to determine the success rate of real threats whenever the scalar sensors trigger the opening of the camera on the sensor node; thus, the sensor node starts object extraction and classification. As a result of these experiments, we understand the impact of both the internal environment and external environment separately under different external conditions.

The success rate is 83%, when the test is done at the indoor conditions, and it is 68%, for the case that the sensor nodes operate in the external environment. The main factor that decreases the success rate of the test is the passing objects like vehicles. The vehicles are already out of the field of view of the camera when scalar sensors activate the camera. The snapshots belong to both successful detection and wasted camera triggering are given in Fig. 5.3.

5.3 Delays of video processing operations

Another experiment that we have conducted on our sensor nodes is to measure time required at each phase of the object detection and classification process. In this experiment we measure the clock times starting from triggering the scalar sensors to determining category of objects. These measurements are carried out on the sensor nodes by using model 4, since it is the most accurate mode according to our experiments. The results of the delay measurements are shown in Table 5.3. Based on these results, we observe that the most time-consuming module is the post-processing operations, which consists of cleaning and sharpening operations of the foreground image.

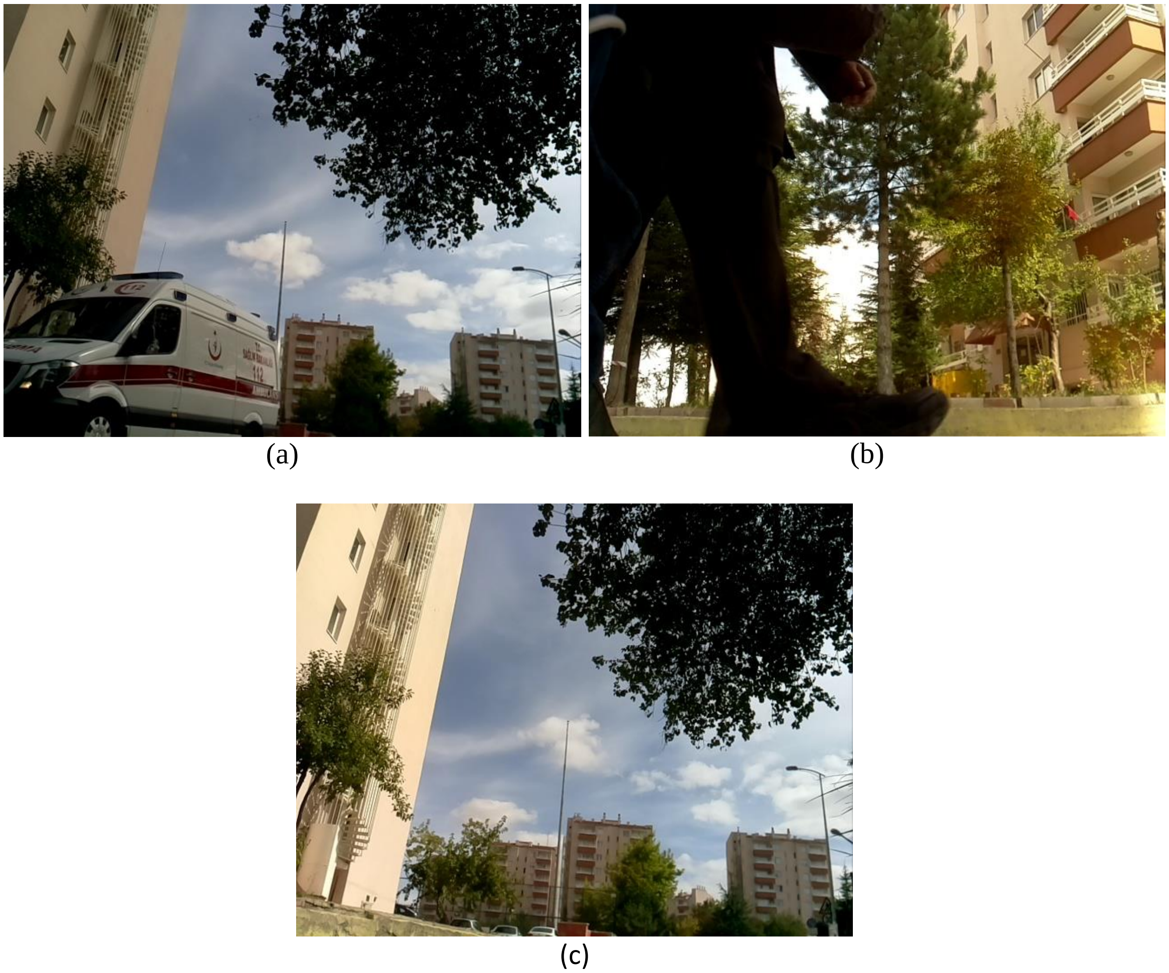

(a)
(b)
(c)

Figure 5.3 Snapshots of tests to evaluate camera triggering success a)Successful triggering by a vehicle, b)Successful triggering by a person, c)Wasted triggering by a passing vehicle

Table 5.3 Sensor Node Operations Durations

Operation	**Duration (sn)**
Opening camera.	0.02
Capture frame	0.02
BS	0.40
Post Processing	0.60
Segmentation	0.01
Classification	0.03

5.4 Audio effect on classification

As it is stated in the previous chapter the sensor node equipped with a second multimedia sensor, microphone. In this part of the experiments we perform tests to see the contribution of the audio based categorization to object classification. While doing that we again use model 4 as method for object categorization from video. During the tests we prefer unified object category for person and group of people so that two categories of objects are to be detected, person/group and vehicle. We apply a high level fusion which fuses audio categorization result and video categorization result by using a decision function. The decision function is given in the formula below;

$$d(\vec{v}_n, a_n) = \begin{cases} \vec{v}_n + a_n & \textit{when } a_n \textit{ is vehicle or person} \\ \vec{v}_n & \textit{other} \end{cases} \tag{5.1}$$

where $\vec{v}_n$ is the list of object categories detected after video processing of frame *n* and a_n is the category detected after audio processing.

A training dataset for audio is constructed in CSV file format. Each audio category has its CSV file, as it is stated before. The contents of CSV files are list of 13 MFCC feature vectors of the audio recordings of the category. As audio file for test video, a separate CSV file is created. In this file there are MFCC vectors which are numbered with the frame numbers of the video. Those MFCC vectors are cut from the training dataset. During video processing the MFCC vector of the processed frame is read from the file and that vector is sent to SVM to detect its category. Then according to the categorization result it is either added to the detected object categories list or discarded.

The results of the experiment are quite satisfying as shown in Table 5.4. We have achieved much better precision/recall rates especially for unified person/group category. We consider that this experiment is proof of contribution of fusing audio based and video based categorization to our object detection and classification approach.

Table 5.3 Comparison of Classification Performance with and without Audio

	Person/Group (Recall/Precision)	**Vehicle (Recall/Precision)**
Without Audio	0,58 / 0,77	0,92 / 0,93
With Audio	0,77 / 0,87	0,99 / 0,90

5.5 Sensor Node Power Consumption

Sensor node power consumption was also tested by using a 3.7-V 1326-mAh Li-ion battery, which is used in mobile telephones. The main goal is to measure the power usage of the sensor node working in various modes and to which mode was the most energy efficient. Before testing the power consumption, current values of the node were measured under multiple conditions; the values are listed in Table 5.5.

Power consumption is compared between the following sensor working modes:
1. Streaming via Wi-Fi: The sensor node captures video and streams it continuously using its Wi-Fi adaptor. The node is directly connected to a workstation’s Wi-Fi adapter. The capturing and streaming process that kept the camera on is started by the sensor node. The video resolution is 160 x 120, the rate is 5 fps, and encoding is H264. Streaming is performed using the VLC application on the sensor node, using the real-time streaming protocol (RTSP). From the workstation VLC is used to capture the streamed video.

2. Our Application using Wi-Fi: In this mode the sensor node application is executed. Communication with the workstation is also carried out over Wi-Fi. Rather than keeping the camera on continuously, it is turned on when a persistent alarm (motion/sound/vibration) is detected by the scalar sensors of the node. Captured frames are processed and the semantic result is sent to the workstation using the Wi-Fi interface. The camera is turned off when the application gets low signals from the scalar sensors.

3. Our Application using ZigBee: The only difference between this mode and the

previous mode is the communication interface. The sensor sends its events to the gateway sensor node using ZigBee interfaces. Different from the previous mode, the camera is not used in this mode. When an object is detected by a scalar sensor, the camera is not turned on for object detection and classification process, but only an event message implying an object detection is transferred through the ZigBee interface.

4. Detection only by scalar sensors: In this scenario we do not use any camera, which is different from the previous mode, mode 3. When an object is detected by a scalar sensor, the camera is not turned on for object detection and classification, but only an event message implying object detection is transmitted through the ZigBee interface.

Table 5.5 Sensor Node Current Measurement

Condition	**Measured Current**
No communication device. No application.	179 mA
Ethernet only connected	220 mA (+41 mA)
Wi-Fi only connected	250-280 mA (+70mA-+100mA)
ZigBee only connected	208 mA (+29 mA)
Camera on	279 mA (+100mA)
Streaming via VLC (160 x 120 resolution,5fps, H264)	245 mA (+65mA)

In order to operate the sensor node with the 3.7-V battery properly, a DC-DC voltage step up regulator is used. The regulator adjusts the input voltage so that the output voltage is fixed at 4.2 V, which is sufficient for the apparatus to work consistently. The step-up regulator uses an 80-mA current to operate, as this current is fixed for all of the working modes, it has no effect on the test results. The test bed is shown in Fig. 5.4. The sensor node stops operation below 2.79 V. The time for the battery to reach its cut-off voltage of 2.79V in each working mode is measured. The tests results are shown in Fig. 5.5.

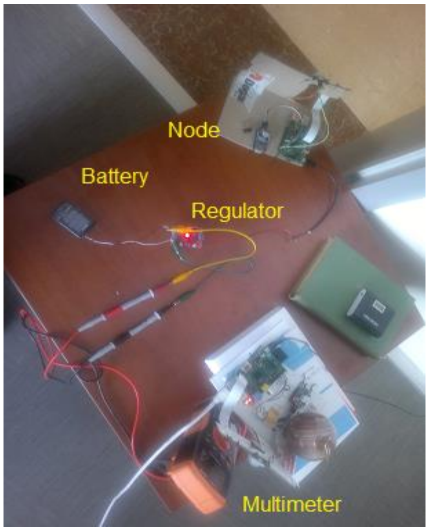


Figure 5.4 Power consumption test bed

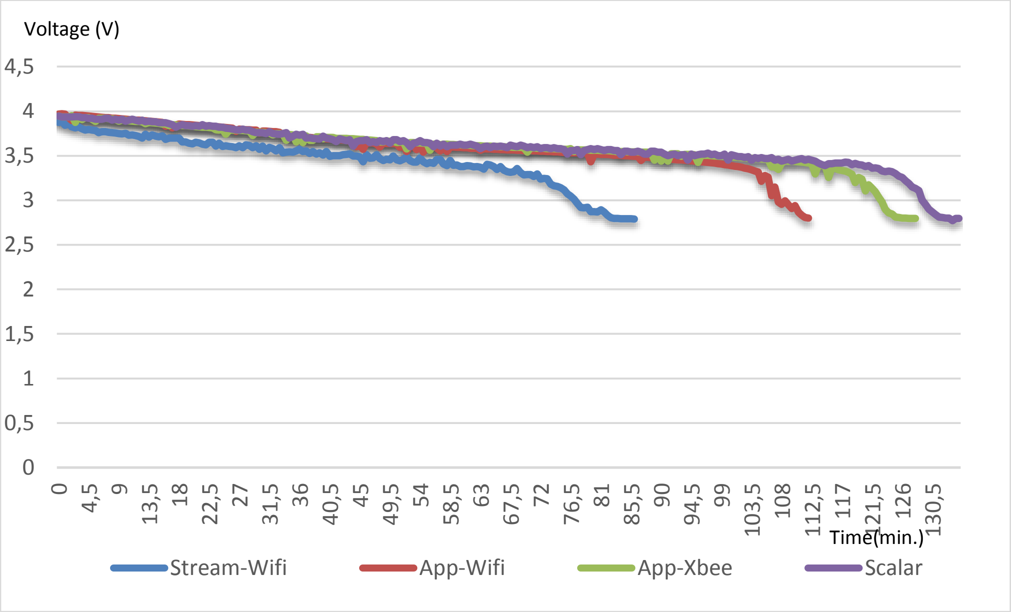


Figure 5.5 Power consumption test results

According to the results, the battery reaches the cut-off voltage earliest (85 min) when it continuously streams the video via Wi-Fi. The camera and Wi-Fi combined draws 200 mA of current and streaming continuously draws another 65 mA while the sensor is active. In operating mode 2, in which the sensor node application runs instead of using continuous streaming, there is a 30% increase in the lifetime of the sensor node, we think the main reason for that is that camera capture and Wi-Fi transmission only occurs when the scalar sensors detects a persistent alarm. Unlike the streaming application, the node application also sleeps in the absence of an alarm, which also results in energy saving. Furthermore, as the quantity of forwarded text-based data is much less than streamed video data, the transmission cost of this working mode is lower.

Power consumption is lower for operating mode 3, which uses ZigBee communication instead of Wi-Fi communication. As compared to the 70 mA used by the Wi-Fi device, the 30 mA used by ZigBee represents a 14% power saving, which makes operation mode 3 the most energy efficient mode with respect to the previous modes. At mode 4, where the camera is not used and no information about a threat is extracted, we have an additional energy gain of 12.5% compared to that of mode 3. Being able to extract concept information at mode 3, despite the energy difference, is considered to be very important for automating surveillance applications.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Herein in this dissertation we described a lightweight WMSN architecture capable of threat detection and classification without human operator intervention. Proposed architecture is a multi-tier one that has scalar sensor layer, multimedia layer and a sink layer. The sensor nodes are setup in clusters. The cluster head is responsible for collecting the information from its cluster and forwarding this information to the sink station.

A real sensor node based on Raspberry-Pi platform was designed and an application was developed and embedded into the sensor node for moving object classification. The sensor node is composed by scalar sensors, PIR, acoustic and vibration as well as a medium resolution camera. ZigBee based communication is setup between sensor nodes. The interoperability of the sensor nodes, sink station, and client applications using the proposed standards and methods was proven.

The application on the node use scalar sensors in order to start capturing frames via its camera. Background subtraction is performed to extract the foreground object from the stationary background. After post processing operations over the foreground image, features of the detected object are extracted from the MBR. Those features are shape based features like width/height ratio, compactness, dispersedness, blob area ratio as well as local invariant SURFs. All those image processing operations are performed by the help of OpenCV library.

Several classification approaches are applied on the extracted features. The performance of those classification approaches are tested using videos recorded by the sensor node. Precision and recall values for each approach are calculated. Extracting information from multimedia data in the WMSN in real time is sufficiently accurate, which eliminates the need for human operators to examine large quantities of multimedia data in real time. An XMPP based communication is setup to forward the information to the sink station and then to the operators. XMPP server and client applications are setup and integrated with the sensor network for that purpose. The proposed architecture is also capable of requesting more complex information from the sensor nodes via operators' client applications. Silhouette of the detected object, foreground picture or live snapshot of the environment are those kind of information.

The role of sending only semantic information extracted from multimedia data to extending the lifetime of the sensor node was also tested. The sensor node is run using different modes and the decline in the battery voltage is measured. The present findings show that the proposed methods provide an energy saving more than 40% and prolong the lifetime of the WMSN. In conclusion, we think that the proposed architecture, methods, and processes presented herein can be effectively used to setup lightweight WMSNs that require minimal manpower for operation.

Within the context of this dissertation there are several areas to be worked on in the future. Those areas can be grouped as;

- Improvements on the Scalar Sensor level
- Improvements and modifications on the application layer
- Work to be done to increase the lightweightness
- Network infrastructure improvements.

Following future works are foreseen to support this study.

1. In order to increase the accuracy of the classification the node can be equipped with scalar sensors that provide more meaningful information. In this study scalar sensors are used to detect the existence of the object and activate the

camera. At the beginning, all classification is based on the multimedia data that camera produces. We show the contribution of fusing video based and audio based categorization to object identification during experiments. Similarly integrating a load cell to the sensor node will provide a valuable weight data which at least can be used for a rule based classification of the object. The information coming from those additional scalar sensors can be fused with the video and audio data in order to increase the performance of the classification process.

2. This study can be considered as a proof of concept to use video and audio processing operations on the multimedia sensor node. The sensor node's application can be improved by implementing different and more complex image and audio processing and classification techniques, as well as other type of features. The effects of these modifications on the system's accuracy, performance and as well as sensor's lifetime can be tested to determine the most effective approach.

3. The sensor node is a Raspberry-Pi platform which can be considered as a mini computer with rich resources with respect to traditional sensor networks. Proposed approach can be applied on a more resource-poor platform like Arduino which has a microcontroller rather than microprocessor and limited memory and storage modules. The complexities of the detection and classification methods are rearranged so that it is possible to execute them on such a limited platform.

4. The network infrastructure of the architecture can also be improved. Without changing the ZigBee infrastructure and without using additional interfaces with high power consumption, several effective and efficient upper layer protocols for MAC and packet routing can be applied and tested. The size of the network in this context can be enlarged and loss of information during data transfer due to

communication failures or weakness of the layer-2 and 3 protocols can be calculated. Besides, transferring imaging data or even streaming live video over such a low bandwidth network can be studied. Effects of different data compression techniques on this kind of transfer can be experimented as well. Buffering mechanisms on the sensor nodes can be improved and tested in order to decrease loss of information due to network failures.

5. Creation of a dataset for future studies on the domain is also a critical task. The sensor nodes can be setup on a real outdoor environment. The multimedia and scalar data that is produced by the sensors can be collected. This data is then annotated and presented to the use of community.

6. Additional functionalities can be implemented at the sink station based on the collected data. It can easily be foreseen that huge amount of data will be stored at the sink server according to the size of the network. Except data types proposed in this dissertation, additional types of information may also be transferred from the network, like extracted feature vectors, scalar information etc. The sink itself is aware of the whole network by the help of that information. Since sink station is much powerful than the sensor nodes that information can be processed exhaustively at the sink to make further decisions like event detection, localization, tracking etc. Big data processing techniques and data mining approaches can be applied on the sink to conclude more accurate and fine results.

Another important work to be done is simulating the network with large number of nodes. The behavior of the network under different circumstances can be observed by modeling the network on a simulation tool like NS. This work has a crucial importance for future improvements not only on the sensor node and application but also on the network architecture.

REFERENCES

[1] A.Sharif, V.Potdar, and E.Chang. "Wireless multimedia sensor network technology: a survey", *in Proc. of the 6th International Conference on Industrial Informatics*, pp. 606-613, Jun. 2009.

[2] I.F.Akyildiz, Y.Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, vol.38, no.4, pp.393-422, 2002.

[3] S.Hengstler, D.Prashanth, S. Fong, and H.Aghajan, "MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance", *in Proc. of the 6th International Conference on Information Processing in Sensor Networks*, pp. 360-369, Apr. 2007.

[4] A Rowe, A.G Goode, D.Goel, and I.Nourbakhsh. (2007, May.) CMUcam3:An: An open programmable embedded vision sensor. Robotics Institute, Carnegie Mellon Univ. [Online].http://repository.cmu.edu /cgi/viewcontent.cgi?article=1849&context=robotics.

[5] A Kandhalu, A.Rowe, and R.Rajkumar, "DSPcam: A camera sensor system for surveillance networks", *in Proc. of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1-7, Aug. 2009.

[6] R.Pon, M.A.Batalin, J.Gordon, A.Kansal, D.Liu, M.Rahimi, L.Shirachi, Y.Yu, M.Hansen, W.J.Kaiser, M.Srivastava, G.Sukhatme, and D.Estrin, "Networked info mechanical systems: a mobile embedded networked sensor platform*", in Proc. of the 4th International Symposium on Information Processing in Sensor Networks*, pp. 376-381, Apr. 2005.

[7] D.Lymberopoulos, A.Savvides, "XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications*", in Proc. of the 4th International Symposium on Information Processing in Sensor Networks*, pp. 449-454, Apr. 2005.

[8] T.Fan, L.Xu, X.Zhang, and H.Wang, "Research and design of a node of wireless multimedia sensor network", *in Proc. of the 5th International Conference on Wireless Communications Networking and Mobile Computing*, pp. 1-5, Sept. 2009.

[9] V.Jelii, M.Magno, D.Brunelli,V.Bilas, and L.Benini, "An energy efficient multimodal wireless video sensor network with eZ430-RF2500 modules", *in Proc. of the 5th International Conference on Pervasive Computing and Applications*, pp. 161-166, Dec. 2010.

[10] J.Campbell, P.B.Gibbons, S.Nath, P.Pillai, S.Seshan, and R.Sukthankar, "IrisNet: an internet-scale architecture for multimedia sensors", *in Proc. of the 13th ACM Int. Conference on Multimedia*, pp. 81-88, Jun. 2005.

[11] P.Kulkarni, D.Ganesan, P.Shenoy, and Q.Lu, "SensEye: a multi-tier camera sensor network", *in Proc. of the 13th ACM International Conference on Multimedia,* pp 229-238, Jun. 2005.

[12] W.Chen, P.Y.Chen, W.S.Lee, and C.F.Huang, "Design and implementation of a real time video surveillance system with wireless sensor networks", *in Proc of the Vehicular Technology Conference,* pp 218-222, May. 2008.

[13] Y.Ur Rehman, M.Tariq, and T.Sato, "A novel energy efficient object detection and image transmission approach for wireless multimedia sensor networks", *IEEE Sensors Journal*, pp. 1, Jun. 2016.

[14] Z.Sun, P.Wang, M.C.Vuran, M.A.Al-Rodhaan, A.M.Al-Dhelaan, and I.F.Akyildiz, "BorderSense: Border patrol through advanced wireless sensor networks", *Ad Hoc Networks,* vol. 9, no. 3, pp. 468-477, May. 2011.

[15] H.Oztarak, K.Akkaya, and A.Yazici, "Lightweight object localization with a single camera in wireless multimedia sensor networks", *in Proc. of the Global Telecommunications Conference,* pp. 1-6, Nov. 2009.

[16] S.Boragno, B.Boghossian, D.Makris, and S.Velastin, "Object classification for real-time video-surveillance applications", *in Proc of the 5th Int. Conference on Visual Information Engineering,* pp. 192-197, Jul. 2008.

[17] M.T Razali, B.J.Adznan, "Detection and classification of moving object for smart vision sensor", *in Proc. of the 2nd International Conference on Information and Communication Technologies*, vol. 1, pp. 733-737, 2006.

[18] G.Bagus, B.Nugraha, S.Weng, and H.Morita, "Multiple object tracking on static surveillance video using field-based prediction information in mpeg-2 video", *in Proc. of the 17th IEEE International Conference on Image Processing*, pp. 4625-4628, Sept. 2010.

[19] H.Y.Lin, J.Y.Wei, "A street scene surveillance system for moving object detection, tracking and classification", *in Proc. of the IEEE Intelligent Vehicles Symposium,* pp. 1077-1082, Jun. 2007.

[20] S.Zhang, C.Wang, S.C.Chan, X.Wei, and C.H.Ho, "New object detection, tracking, and recognition approaches for video surveillance over camera network", *IEEE Sensors Journal,* vol. 15, no. 5, pp. 2679-2691, Mar. 2015.

[21] L.M.Brown, “View independent vehicle/person classification”, *in Proc.of the 2nd ACM International Workshop on Video Surveillance & Sensor Networks*, pp. 114-123, Oct. 2004.

[22] M.Chitnis, Y.Liang, J.Y.Zheng, P.Pagano, and G.Lipar, “Wireless line sensor network for distributed visual surveillance”, *in Proc. of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pp. 71-78, Oct. 2009.

[23] M.O.Farooq and T.Kunz, “Wireless Multimedia Sensor Networks Testbeds and State-of-the-Art Hardware: A Survey”, *in Communication and Networking Springer Berlin Heidelberg*, pp. 1–14, 2011.

[24] I.T.Almalkawi, M.G.Zapata, J.N.Al-Karaki, and J.M.Pozo, “Wireless multimedia sensor networks: Current trends and future directions”, *Sensors*, vol. 10, no. 7, pp. 6662-6717, Jul. 2010.

[25] J.Yick, B.Mukherjee, and D.Ghosal, “Wireless sensor network survey”, *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, Aug. 2008.

[26] Ian F. Akyildiz, T.Melodia, Kaushik R. Chowdhury, “A survey on wireless multimedia sensor networks”, *Computer Networks*, vol. 51, pp. 921–960, 2007.

[27] W.Li, X. Wu, K.Matsumoto, and H.A.Zhao, “Foreground detection based on optical flow and background subtraction”, *in Proc. of the Int. Conference on Communications, Circuits and Systems*, pp. 359-362, Jul. 2010.

[28] Y.Benezeth, P.M.Jodoin , B.Emile, H.Laurent, and C.Rosenberger, “Comparative study of background subtraction algorithms”, *Journal of Electronic Imaging*, vol. 19, no. 3, Jul. 2010.

[29] C.Stauffer, W.E.L.Grimson, “Adaptive background mixture models for real-time tracking”, *in Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Jun. 1999.

[30] M.Piccardi, “Background subtraction techniques: a review”, *in Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099-3104, Oct. 2004.

[31] R.M.Haralick, S.R.Sternberg, and X.Zhuang, “Image analysis using mathematical morphology”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532-550, Jul. 1987.

[32] R. H.Chan, C.WaHo, and M.Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization", *IEEE Trans. on Image Processing*, vol. 14, no. 10, pp. 1479-1485, Oct. 2005.

[33] N.Senthilkumaran, R.Rajesh, "Edge detection techniques for image segmentation–a survey of soft computing approaches", *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, Nov. 2009.

[34] M.B.Dillencourt, H.Samet, and M.Tamminen, "A general approach to connected-component labeling for arbitrary image representations", *Journal of the Association for Computing Machinery*, vol. 39, no. 2, pp. 253-280, Apr. 1992.

[35] T.Tuytelaars, K.Mikolajczyk, "Local invariant feature detectors: A survey", *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177-280, Jan. 2008.

[36] D.G.Lowe, "Object recognition from local scale-invariant features", *in Proc. of the 7th IEEE International Conference on Computer Vision*, vol. 2, pp. 1150-1157, Sep. 1999.

[37] H.Bay, T. Tuytelaars, and L.V.Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, Jun. 2008.

[38] P.Viola, M.Jones, "Rapid object detection using a boosted cascade of simple features*", in Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.

[39] N.Dalal, B.Triggs, "Histograms of oriented gradients for human detection", *in Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, Jun. 2005.

[40] D. Lu, Q. Weng, "A survey of image classification methods and techniques for improving classification performance*", International Journal of Remote Sensing*, vol. 28, no. 5, pp 823-870, Mar. 2007.

[41] Text book, "Remote Sensing Notes", *Japan Association of Remote Sensing (JARS),* 1999.

[42] C.Cortes, V.Vapnik, "Support vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273-297, Sep. 1995.

[43] *Wikipedia The Free Encyclopedia*. [Online]. Available:
https://en.wikipedia.org/wiki/Support_vector_machine, accessed Nov. 2016.

[44] M.Hofmann, “Support Vector Machines—Kernels and the Kernel Trick”, Notes , 2006.

[45] *CMU Computer Vision Lecture Notes on Bag of Words, K.Kitani*. [Online]. Available: http://www.cs.cmu.edu/~16385/lectures/Lecture12.pdf, accessed: Nov. 2016.

[46] S.Adolph, M.Reisslein. “Towards efficient wireless video sensor networks: A survey of existing node architectures and proposal for a Flexi-WVSNP design”, *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 462-486, Sep. 2011.

[47] M.Rahimi, R.Baer, J.Warrior, D.Estrin, and M.Srivastava. “Cyclops: in situ image sensing and interpretation in wireless sensor networks”, *in Proc. of the 3rd Int. Conf. on Embedded Networked Sensor Systems*, pp. 192-204, Nov. 2005.

[48] Stargate platform. Available:http://www.xbow.com/Products/XScale.htm

[49] A.J.Lipton, H.Fujiyoshi, R.S.Patil , “Moving Target Classification and Tracking from Real-time Video”, *in Proceedings Fourth IEEE Workshop on Applications of Computer Vision*, pp. 8-14, 1998.

[50] T.Damarla, A.Mehmood, and J.Sabatier, “Detection of people and animals using non-imaging sensors”, *in Proc. of the 14th International Conference on Information Fusion*, pp. 1-8, Jul. 2011.

[51] A.Clemens, B.Horst, and L.Christian, “Tricam-an embedded platform for remote traffic surveillance”, *in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 125, Jun. 2006.

[52] M.Chitnis, Y.Liang, J.Zheng, P.Pagano, and G.Lipari, “Wireless line sensor network for distributed visual surveillance”, *in Proc. of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pp. 125, Oct. 2009.

[53] H.Oztarak, T.Yılmaz, K.Akkaya, and A.Yazici, “Efficient and accurate object classification in wireless multimedia sensor networks”, *in Proc. of the 21st Int. Conf. on Computer Communication Networks*, pp. 1-7, 2012.

[54] D.Kim, S.Rho, E.Hwang, “Local feature-based multi-object recognition scheme for surveillance”, *in Engineering Applications of Artificial Intelligence* , vol. 25, no. 7, pp. 1373–1380, 2012.

[55] *OpenCV open source computer vision library*. [Online]. Available: http://opencv.org, accessed Nov. 2015.

[56] *Gloox full-featured Jabber/XMPP client library*. [Online]. Available: http://camaya.net/gloox/, accessed Dec. 2015.

[57] *wiringPi GPIO access library*. [Online]. Available: http://wiringpi.com/, accessed Sep. 2015 .

[58] *Ignite Realtime site for realtime communication projects*. [Online]. Available: http://www.igniterealtime.org, accessed Nov. 2015.

[59] *XMPP Standards Foundation*. [Online]. Available: http://xmpp.org/, accessed Dec. 2015.

[60] *Sensarea authoring tool for video object editing.* [Online]. Available: http://www.gipsa-lab.grenoble-inp.fr/~pascal.bertolino/projets.html, accessed Feb. 2016.

VITA

Muhsin Civelek received his BSc degree in Computer Science Engineering from Marmara University in 1998 and MSc degree in Software Management from Middle East Technical University in 2006. He worked as a wide area network administrator and telecommunication network administrator for eleven years. Then he worked as an instructor in Computer Engineering Department of Turkish Military Academy. He gave Computer Network courses in that department. His research interests include computer networks, sensor networks and embedded programming.

Publications

1. M.Civelek, A.Yazıcı, "Automated Moving Object Classification in Wireless Multimedia Sensor Networks", IEEE Sensors Journal, 2016.

2. M.Civelek, A.Yazıcı, C.Yılmazer and F.Ö.Korkut, "Feature extraction and object classification for target identification at wireless multimedia sensor networks", 22nd Signal Processing and Communications Applications Conference (SIU). IEEE, 2014.

3. M.Civelek, A.Yazici., "Object Extraction and Classification in Video Surveillance Applications", European Review 1-14, 2016.