

ANALYSIS OF SPECTRUM SENSING USING MACHINE LEARNING

Submitted in partial fulfillment of the
requirements for the award of

Bachelor of Engineering in Electronics and Communication Engineering

By

A SAHIT (39130036)

K SHANMUKHA SAI (39130200)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —All grade by NAAC Jeppiaar
Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Final Year Project Report is the bonafide work of **ARISSETTY SAHIT (39130036)** and **KALUGURI SHANMUKHA SAI (39130200)** who have done the project work as an team who carried out the project entitled “**ANALYSIS OF SPECTRUM SENSING USING MACHINE LEARNING**” under our supervision from JULY 2022 to April 2023.

Internal Guide

MS G. I SHAMINI, M.Tech, (Ph.D).,

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **A. Sahit** (39130036) and **K. Shanmukha Sai** (39130200) hereby declare that the Project Report entitled “**Analysis of Spectrum Sensing Using Machine Learning**” done by us under the guidance of **Ms G. I SHAMINI, M.Tech, (Ph.D)** Assistant Professor at Sathyabama institute of Science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 26.04.2023

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

1.



2.



ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Professor & Dean,** School of Electrical and Electronics, **Dr. T. RAVI, M.E., Ph.D., Professor & Head,** Department of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to our **Project Guide Ms G.I SHAMINI, M.Tech, (Ph.D)** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Spectrum sensing is a key function of cognitive radio to prevent the harmful interference with licensed users and identify the available spectrum for improving the spectrum's utilization. The limited available spectrum of satellite and space communications result in new techniques to reuse the free spectrum. Cognitive Radio (CR) is one of the most promising techniques for such reusing processes. Spectrum sensing (SS) is the core process of cognitive radio which is used to sense the available temporary free bands, holes, of the sensed spectrum. Many technologies are proposed to achieve both narrow-band and wide-band spectrum sensing. Many scenarios are proposed to utilize cognitive radio architecture at both standalone and cooperative approaches in satellite communications. The main aim to propose a scenario that efficiently saves frequency resources, noise and interference problems with an accurate reconstruction. In this paper, we consider the application of machine learning (ML) models in cooperative spectrum sensing (SS) of cognitive radio networks (CRNs) to cancel the noise.

TABLE OF CONTENTS

Chapter No	TITLE	Page No
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Problem statement	3
	1.3 Objective	3
2	LITERATURE SURVEY	4
	2.1 Machine Learning Meets Point Process: Spatial Spectrum Sensing in User-Centric Networks	4
	2.2 Technical issues on cognitive radio-based Internet of Things systems: A survey	4
	2.3 Generalized energy detection under generalized noise channels	5
	2.4 Machine Learning-Enabled Cooperative Spectrum Sensing for Non-Orthogonal Multiple Access	6
3	SYSTEM DESIGN	7
	3.1 Description of Existing Method	7
	3.2 Block design and its explanation	7
4	DESCRIPTION OF PROPOSED SYSTEM	9
	4.1 Proposed Methodology	9
	4.2 Block Design of Proposed System	9
	4.3 Description of Software for Implementation	10
5	IMPLEMENTATION DETAILS	11
	5.1 Flowchart implementation	11

	5.2	Module development and description	15
	5.3	Development and Deployment	17
6		SOFTWARE DESCRIPTION	32
7		CONCLUSION	35
		REFERENCES	39
		APPENDIX	41
	A	SOURCE CODE	41

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1.1	Cognitive Radio	2
3.2.1	Block diagram to remove Noise	7
4.2.1	Block diagram of energy detection	9
5.1.1	Flow Chart of Sensing Technique	11
5.3.1	Support vector machine block diagram	20
5.3.2	Convolution	21
5.3.3	1st and 2nd derivative of an edge is illustrated in one dimension	24
5.3.4	Simple gray level image and the corresponding surface in image space	27
5.3.5	Two connected components based on 4-connectivity	31
6.1	Complementary ROC of Cooperative Sensing with AND Rule under AWGN	37
6.2	Comparison of Total Error Rates using Energy Detection and Matched Filter Detection	38

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Two decades ago, Mitola introduced a new concept in wireless telecommunication: the Cognitive Radio (CR) [1]. CR is mainly based on Soft Defined Radio (SDR) [2], where specific hardware can be replaced by more generic hardware that can be configured via software. In addition to being softly configurable, CR is aware and adaptable to the radio environment, which can be exploited in optimizing the use of available frequency bands while protecting the occupied ones from harmful interference. Most of the current wireless communication systems are based on the concept of fixed (or static) frequency allocation. They are designed to operate on pre-selected frequency bands. This static allocation results in a low spectrum utilization especially at low traffic periods. Based on [3], the usage of some allocated frequency bands is lower than 15%. In addition, the development towards 5G and Beyond (B5G) technologies, the exponential growth in the number of connected objects via the Internet of Things (IoT), Wireless Sensor Network (WSN) devices and recent wireless applications push the wireless communication.

Community to enhance the utilization of the limited frequency resources to satisfy the increasing demand on the wireless communication services [4,5]. CR has been introduced as a potential candidate to perform complete Dynamic Spectrum Allocation (DSA) by exploiting the free frequency bands that are also called “spectrum holes” or “white spaces” [6–8]. Being capable to identify these spectral opportunities, CR classifies the users into two categories: licensed, i.e., the Primary Users (PUs), and unlicensed, i.e., the Secondary Users (SUs). While PUs can access the spectrum whenever they want, SUs are restricted by the activities of PUs. In other words, SUs should respect the PUs’ Quality of Service and harmful interference coming from SUs to PUs transmission is prohibited. Therefore, three paradigms of CR can be distinguished according to the possibility of co-existence of SU and PU transmissions in the same channel, the permitted transmit power of SU and the cooperation between SU and PU. As shown in Figure 1, three main paradigms of CR can be distinguished: 1. Underlay Access: the SU may

transmit simultaneously with the PU over the same channel. However, the transmitted power should not exceed a certain threshold in order to keep the interference on PU below a tolerable value [4,9]. 2. Overlay Access: the SU may transmit simultaneously with the PU on the same channel up to its maximum power, but at the cost of playing a role of relay between two or more PUs [10,11]. In this case, the SU sends its data while relaying the PUs. 3. Interweave Access: SU is allowed to transmit using its maximum power only when PU is absent. This paradigm is also known as the classical CR and it is the focus of this paper given its popularity. The main drawback of the underlay paradigm is the low transmitted power, which adversely impacts the throughput. The use of the overlay paradigm is limited to scenarios where PU and SU have a high level of cooperation. Interweave paradigm allows SUs to transmit with their maximum power, but at the cost of monitoring the activity of PU. In classical interweave systems, the SU activity period is divided into two time slots: sensing and transmission. This leads to the so-called Half-Duplex Cognitive Radio (HDCR).

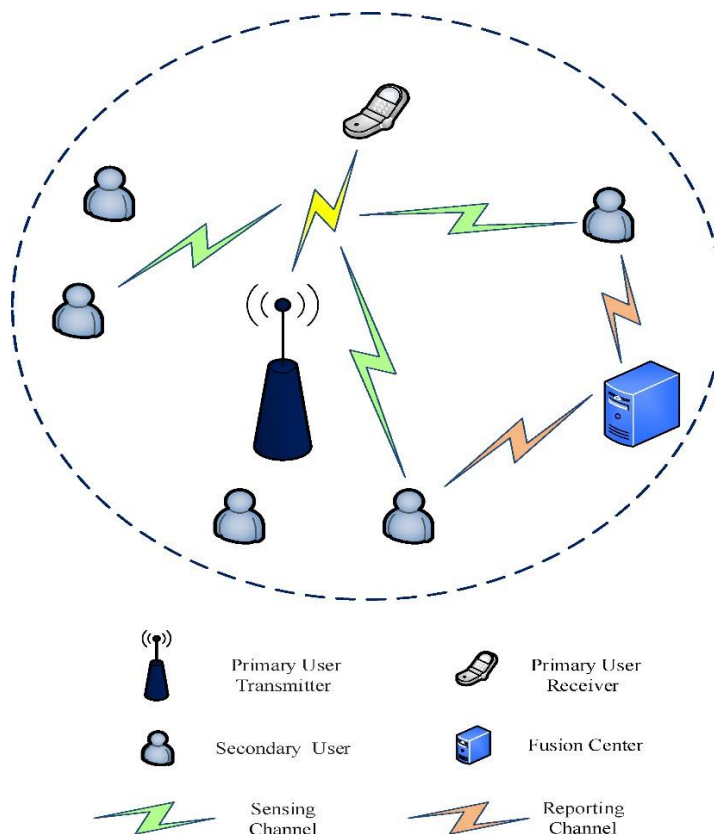


Fig. 1.1.1 Cognitive Radio

CR technology has the potential to provide significant benefits in terms of spectral efficiency and usage. However, there are still several technical and regulatory challenges that need to be addressed before widespread adoption can occur. One of the main technical challenges is the design of reliable sensing techniques that can accurately detect PU activities while minimizing the false alarm rate. Another challenge is the development of efficient and robust spectrum sharing protocols that can enable SUs to access the spectrum dynamically without causing harmful interference to PUs. In addition, regulatory issues related to spectrum management and allocation, as well as the protection of PUs from harmful interference, need to be addressed. Despite these challenges, CR technology has attracted significant research attention and is expected to play a key role in future wireless communication systems.

1.2 PROBLEM STATEMENT

- Spectrum sensing is a key function of cognitive radio to prevent the harmful interference with licensed users and identify the available spectrum for improving the spectrum's utilization.
- In this world, Noise is the major problem for Communication from one place to another using radio signals.
- There are many techniques to decrease the noise. In the one of the best techniques is Active noise control system.
- But that can only decrease at 60% of noise and its is too expensive, Hard to use.

1.3. OBJECTIVE

- To develop efficient model for cancelling the noise using cognitive radio by machine learning model.
- By considering and predicting best efficient model using machine learning is best technique.
- Our primary objective is to develop a model which gives more than 80 to 90% of efficiency in noise cancellation for communication.

CHAPTER 2

LITERATURE SURVEY

2.1 MACHINE LEARNING MEETS POINT PROCESS: SPATIAL SPECTRUM SENSING IN SER-CENTRIC NETWORKS

This letter introduces a novel machine learning (ML)-based approach to approximate the distributions of the aggregated interference power in wireless networks. We focus on the application of spatial spectrum sensing (SSS) in user-centric networks where Poisson cluster process (PCP) is used to model the primary users. A nonlinear regression method, i.e., kernel regression, is introduced to learn the distributions of the aggregated interference power of the PCP modeled primary user network. Simulation results demonstrate the accuracy of our approach.

This letter focuses on the SSS in PCP modeled primary user's networks and discusses the technical issues on obtaining the distributions of the aggregated interference power analytically. Furthermore, a ML-based approach to approximate the distributions are introduced. Simulation results verify the accuracy of the proposed approach. In future work, similar approaches are encouraged to be applied to various realistic scenarios that may have more complex models, such as unmanned aerial vehicle (UAV) networks, vehicle-to-everything (V2X) networks, and heterogeneous user-centric deployed cellular networks, etc. The proposed ML-based approach provides a flexible and efficient way to approximate the distributions of the aggregated interference power in user-centric networks.

2.2 TECHNICAL ISSUES ON COGNITIVE RADIO-BASED INTERNET OF THINGS SYSTEMS: A SURVEY

Cognitive radio (CR)-based Internet of Things (IoT) system is an effective step toward a world of smart technology. Many frameworks have been proposed to build CR-based IoT systems. The CR-based IoT frameworks are the key points on which this survey focuses. Efficient spectrum sensing and sharing are the main functional components of the CR-based IoT. Reviews of recent SS and sharing approaches are presented in this survey.

This survey classifies the SS and sharing approaches and discusses the merits and limitations of those approaches. Moreover, this survey discusses the design factors of the CR-based IoT and the criteria by which the proper SS and access approaches are selected. Furthermore, the survey explores the integration of newly emerging technologies with the CR-based IoT systems. Finally, the survey highlights some emerging challenges and concludes with suggesting future research directions and open issues.

The demand for smart and CRN enabled IoT systems is ever increasing to perceive the ambient radio conditions, analyze RF spectrum and make decisions how, when and where to transmit. This demand led to extensive research on designing CR-based IoT systems as typical paradigm for smart IoT systems by incorporating the cognition capabilities of CR. This survey has provided the existing frameworks of CR based IoT systems, explored the recent SS and spectrum sharing approaches and highlighted the advantages and disadvantages of each approach. Moreover, the survey has identified the essential requirements of constructing CR-based IoT systems and highlighted the design factors. Furthermore, the elements of cognition cycle of CR-based IoT systems had been presented and discussed. By using kernel regression, the approach is able to learn the nonlinear relationships between the interference power and the relevant features.

2.3 GENERALIZED ENERGY DETECTION UNDER GENERALIZED NOISE CHANNEL

Generalized energy detection (GED) is analytically studied when operates under fast-faded channels and in the presence of generalized noise. For the first time, the McLeish distribution is used to model the underlying noise, which is suitable for both non-Gaussian (impulsive) as well as classical Gaussian noise channels. Important performance metrics are presented in closed forms, such as the false-alarm and detection probabilities as well as the decision threshold. Analytical and simulation results are cross-compared validating the accuracy of the proposed approach in the entire signal-to-noise ratio regime. Finally, useful outcomes are extracted with respect to GED system settings under versatile noise environments and when noise uncertainty is present.

This paper presents a study on the performance of generalized energy detection (GED) in the presence of generalized noise and fast-faded channels. The McLeish distribution is introduced to model both non-Gaussian and Gaussian noise channels. Performance metrics such as false-alarm and detection probabilities, as well as decision thresholds, are presented in closed forms. Analytical and simulation results are compared to validate the accuracy of the proposed approach in different signal-to-noise ratio regimes. The study provides useful insights on GED system settings under versatile noise environments and when noise uncertainty is present.

2.4 MACHINE LEARNING-ENABLED COOPARATIVE SPECTRUM SENSING FOR NON-ORTHOGONAL MULTIPLE ACCESS

In this paper, multiple machine learning-enabled solutions are adopted to tackle the challenges of complex sensing model in cooperative spectrum sensing for non-orthogonal multiple access transmission mechanism, including unsupervised learning algorithms (K-Means clustering and Gaussian mixture model) as well as supervised learning algorithms (directed acyclic graph-support vector machine, K-nearest-neighbor and back-propagation neural network). In these solutions, multiple secondary users (SUs) collaborate to perceive the presence of primary users (PUs), and the state of each PU need to be detected precisely.

In this paper, multiple machine learning-enabled solutions (including K-Means, GMM, DAG-SVM, KNN and BP neural network) were introduced to solve the CSS problem for NOMA. Extensive numerical results showed that the proposed solutions can achieve accurate as well as effective spectrum sensing and DAG-SVM algorithm was more suitable than other algorithms. Moreover, we found that there was an optimal power allocation between the two PUs and the proposed solutions can achieve a well sensing accuracy in the case of poor SNR at the receivers. Additionally, the proposed solutions were evaluated for different scenarios, such as varying number of SUs, different training data volume, and different power allocation between PUs. The results show that the performance of the proposed solutions is robust and can adapt to various scenarios. This work can be extended to other non-orthogonal multiple access transmission mechanisms and can be applied in practical cognitive radio systems.

CHAPTER 3

SYSTEM DESIGN

3.1 DESCRIPTION OF EXISTING METHOD

Active Noise Control systems (ANCs), are one of the recommended methods for reducing unwanted noise in a variety of environments, including hospitals. This method is an effective alternative to passive methods based on insulators and dampers

Disadvantages

1. They can be expensive. Noise-canceling headphones can be expensive, especially if you buy a quality pair.
2. They can be difficult to use.
3. They can be distracting.
4. They can be dangerous if not used properly. ..They can be a distraction in social settings.

3.2 BLOCK DESIGN AND ITS EXPLANATION

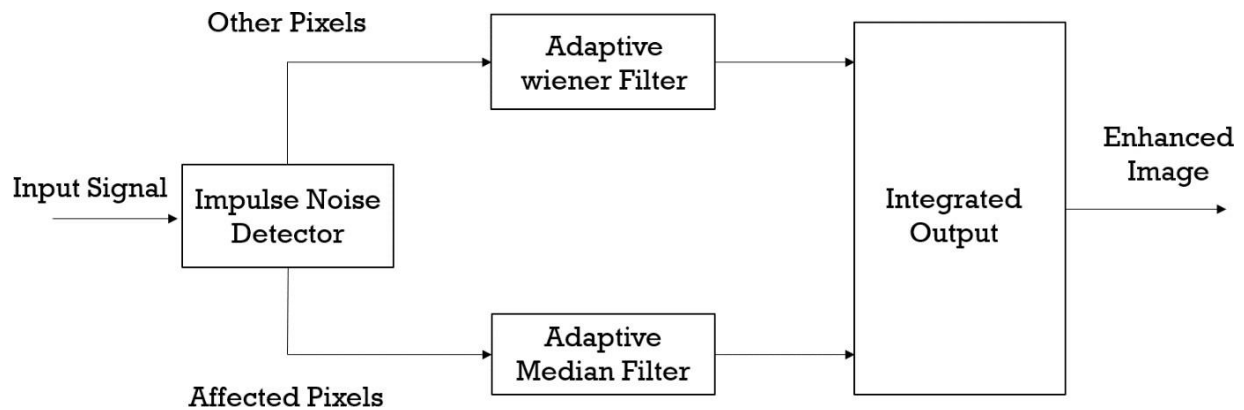


Fig 3.2.1 Block diagram to remove Noise

The circuit consists of an analog-to-digital converter (ADC) that converts an analog input signal into digital form, a digital signal processor (DSP) that filters out unwanted noise from the signal, and a digital-to-analog converter (DAC) that converts the filtered digital signal back to analog form. The purpose of this circuit is to reduce the amount of noise in a signal and improve its quality. The ADC converts the analog signal into a series of digital values, which are processed by the DSP to remove any unwanted noise. The resulting filtered digital signal is then converted back to analog form by the DAC, producing a clean and improved output signal.

This type of noise filter circuit is commonly used in a variety of applications, such as audio processing, image processing, and data communication. The specific implementation of the circuit and the type of filter used can vary depending on the specific application requirements. The circuit takes a noisy signal as input and outputs a filtered signal with reduced noise. The ADC converts the analog signal into a series of digital values, which are processed by the DSP to remove any unwanted noise. The resulting filtered digital signal is then converted back to analog form by the DAC, producing a clean and improved output signal.

The circuit consists of three main components:

1. Analog-to-Digital Converter (ADC): The noisy signal is first converted from analog to digital form using an ADC.
2. Digital Signal Processor (DSP): The digital signal is then processed using a DSP, which applies a filter to remove noise from the signal.
3. Digital-to-Analog Converter (DAC): The filtered digital signal is then converted back to analog form using a DAC, resulting in a clean and filtered analog signal.

Overall, this block diagram represents a basic noise filter circuit that can be used to improve the quality of a noisy signal.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

1. Cognitive Radio (CR) is one of the most promising techniques for such reusing processes.
2. Spectrum sensing (SS) is the core process of cognitive radio which is used to sense the available temporary free bands, holes, of the sensed spectrum.
3. It is highly required to achieve real-time communication for spectrum sensing in cognitive radio due to the intensive required computations.
4. Here we going to separate the noises from the audio signal.

Advantages

1. Overcome radio spectrum scarcity.
2. Avoid intentional radio jamming scenarios.
3. Switch to power saving protocol.
4. Improve satellite communications.
5. Improves quality of service (QoS)

4.2 BLOCK DESIGN

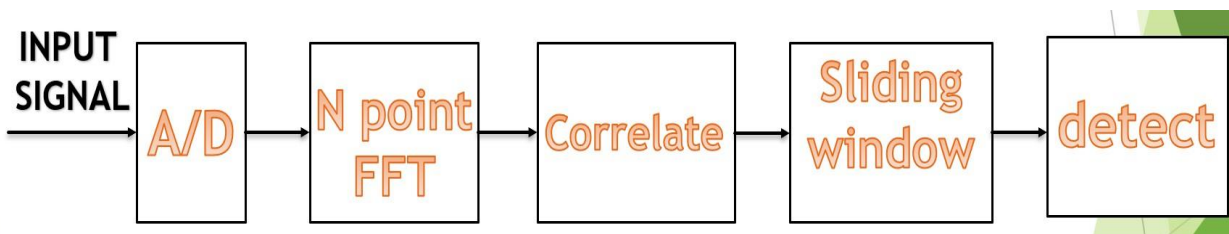


Fig 4.2.1 Block diagram of energy detection

4.3 DESCRIPTION OF SOFTWARE FOR PROPOSED SYSTEM

This circuit aims to enhance the quality of a signal by reducing the amount of noise it contains. It comprises an analog-to-digital converter (ADC) that converts the analog signal into digital form, a digital signal processor (DSP) that filters out any unwanted noise, and a digital-to-analog converter (DAC) that changes the filtered digital signal back to analog form. This type of noise filter is widely used in various applications, such as data communication, image and audio processing.

Here's a breakdown of the different components in the block diagram:

1. **Signal source:** This represents the source of the signal that needs to be detected. It could be a transmitter, a receiver or any other device that emits signals.
2. **Bandpass filter:** This is used to filter out all the signals except for the ones in a specific frequency band. The bandpass filter allows only the frequencies of interest to pass through and attenuates all other frequencies.
3. **Low noise amplifier (LNA):** This component is used to amplify the weak signals that pass through the bandpass filter. The LNA helps to improve the signal-to-noise ratio of the incoming signal, which is important for accurate detection.
4. **Envelope detector:** The envelope detector extracts the envelope of the modulated signal. The envelope is the variation in amplitude of the signal, which is important for detecting the presence of a signal.
5. **Threshold detector:** This component compares the output of the envelope detector with a predefined threshold value. If the output of the envelope detector is greater than the threshold value, the threshold detector outputs a binary signal indicating the presence of a signal.
6. **Controller:** The controller receives the output of the threshold detector and controls any further actions, such as triggering an alarm or initiating a countermeasure.

Overall, the energy detection system is a simple and effective way to detect the presence of signals in a given frequency band. By filtering out unwanted signals and amplifying weak signals, the system can accurately detect the signals of interest and trigger appropriate actions as needed.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 FLOWCHART IMPLEMENTATION:

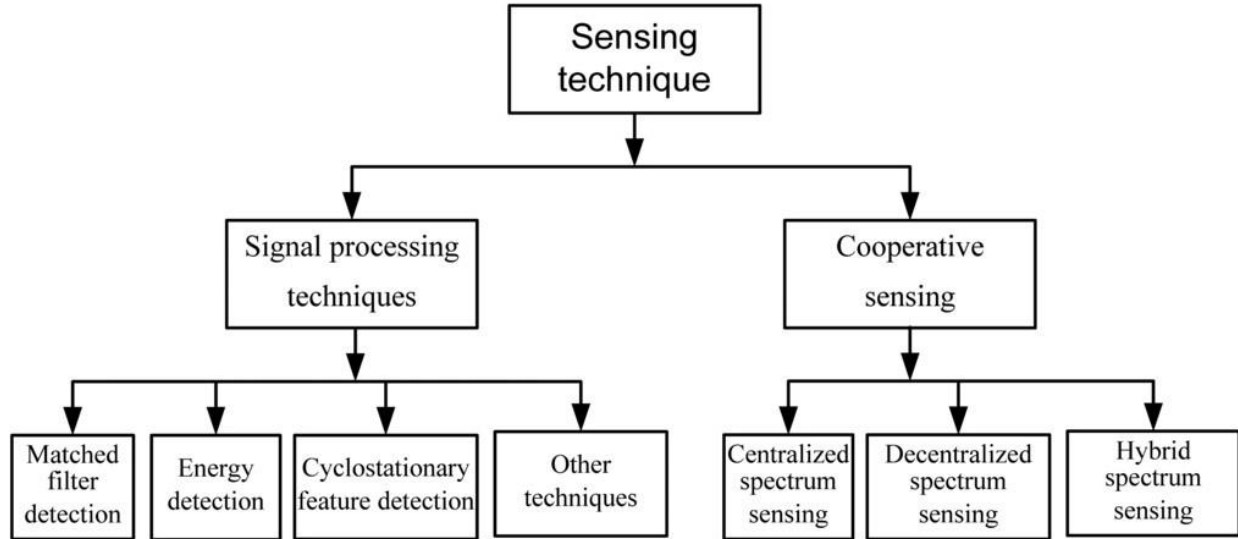


Fig 5.1.1 Flow Chart of Sensing Technique

Spectrum sensing is a technique used in wireless communication systems to detect the presence of signals in a given frequency band. The process begins with signal detection, which is the first step in spectrum sensing. In this step, the receiver scans the frequency band to detect the presence of any signals. The receiver checks if there is any signal present above a certain threshold level. If there is no signal present, the receiver continues to scan the frequency band.

If a signal is detected, the next step is energy detection. This technique measures the energy of the received signal and compares it with a predefined threshold value. If the energy of the received signal exceeds the threshold, it is considered to be present in the frequency band. The advantage of energy detection is that it is a simple technique that does not require knowledge of the transmitted signal. However, it may not be suitable for detecting weak signals in the presence of noise.

Another technique for detecting signals is matched filter detection. This technique uses a matched filter to detect the presence of a known signal in the received signal. The matched filter is designed to maximize the signal-to-noise ratio (SNR) of the received signal. If the output of the matched filter exceeds a certain threshold value, the signal is considered to be present in the frequency band. The advantage of matched filter detection is that it is more robust than energy detection in the presence of noise. However, it requires knowledge of the transmitted signal and may not be suitable for detecting unknown signals.

In summary, spectrum sensing is an essential technique for wireless communication systems to detect the presence of signals in a given frequency band. The flow chart represents the process of spectrum sensing, starting with signal detection and followed by energy detection or matched filter detection. The choice of detection technique depends on the application requirements, signal characteristics, and noise conditions.

Cognitive Radio:

Cognitive radio (CR) is a type of wireless communication system that uses advanced software algorithms to dynamically adapt to the radio frequency spectrum. The main objective of CR is to improve the utilization of the radio frequency spectrum by allowing devices to identify and use available frequency bands that are not being used by other devices.

The fundamental concept of CR is based on the idea that the radio frequency spectrum is a valuable and limited resource that should be used efficiently. In a traditional wireless communication system, the spectrum is statically allocated to specific users, which can result in inefficiencies and underutilization of the spectrum. In contrast, CR systems are designed to be flexible and adaptive, allowing devices to identify available frequency bands and use them as needed.

CR systems consist of several components, including a spectrum sensing module, a decision-making module, a communication module, and a learning module. The spectrum sensing module is responsible for detecting available frequency bands, while the decision-making module uses this information to determine the best frequency band to

use. The communication module is used to transmit and receive data, while the learning module is used to continuously update the system's knowledge of the radio frequency spectrum.

The advantages of CR systems include increased spectrum efficiency, improved quality of service, and reduced interference. CR systems are particularly useful in applications where the radio frequency spectrum is crowded or where spectrum availability changes frequently, such as in military communications, emergency response systems, and public safety networks.

In summary, cognitive radio is a type of wireless communication system that uses advanced software algorithms to dynamically adapt to the radio frequency spectrum. By identifying and using available frequency bands, CR systems can improve spectrum efficiency, improve quality of service, and reduce interference.

Spectrum Sensing:

Spectrum sensing is a technique used in wireless communication systems to detect the presence of signals in a given frequency band. The purpose of spectrum sensing is to identify the available frequency bands that can be used for communication and to avoid interference with other devices operating in the same frequency band.

There are several spectrum sensing techniques available, including energy detection, matched filter detection, cyclostationary feature detection, and wavelet-based detection. Each technique has its advantages and disadvantages, and the choice of technique depends on the specific application requirements and noise conditions.

Spectrum sensing is a critical function in wireless communication systems that allows devices to detect the presence of signals in a given frequency band. The objective of spectrum sensing is to determine the available frequency bands that can be used for communication, and to avoid interference with other devices operating in the same frequency band. The implementation of spectrum sensing involves the use of various techniques which has its own strengths and limitations. Energy detection, matched filter detection, cyclostationary feature detection, and wavelet-based detection are some of the commonly used techniques in spectrum sensing.

Energy Detection :

Energy detection is a technique used in wireless communication systems to detect the presence of signals in a given frequency band. It works by measuring the energy of the received signal and comparing it with a predefined threshold value. If the energy of the received signal exceeds the threshold, it is considered to be present in the frequency band. Energy detection is a simple technique that does not require knowledge of the transmitted signal. It is commonly used in applications where the signal-to-noise ratio (SNR) is high, such as in radar and satellite communication systems. The advantage of energy detection is its simplicity and low computational complexity, making it suitable for implementation in low-cost and low-power devices.

However, energy detection may not be suitable for detecting weak signals in the presence of noise, as it may result in false positives or false negatives. To improve the performance of energy detection, various techniques such as averaging, threshold adaptation, and multiple antenna diversity can be used. Furthermore, energy detection can be combined with other spectrum sensing techniques, such as matched filter detection, to enhance the accuracy of signal detection.

Overall, energy detection is a fundamental technique for spectrum sensing in wireless communication systems, which provides a simple and low-cost approach to detect the presence of signals in a given frequency band. Its implementation depends on the specific application requirements and noise conditions, and its performance can be improved using various techniques.

Cyclostationary Energy:

Cyclostationary energy is a property of a signal that exhibits periodic behavior in its statistical properties over time. This property is observed in signals that have non-stationary features, meaning that the statistical properties of the signal change over time. Cyclostationary energy can be used in spectrum sensing techniques to detect the presence of signals in a given frequency band. Cyclostationary energy is characterized by the cyclic autocorrelation function (CAF), which is a measure of the correlation between a signal and a delayed version of itself. The CAF is periodic in nature, with a

period equal to the reciprocal of the signal's cyclic frequency. The cyclic frequency is a measure of the rate at which the signal's statistical properties vary with time.

To detect cyclostationary energy, cyclic feature detection techniques can be used, such as the cyclic periodogram or the cyclic autocorrelation function. These techniques exploit the cyclostationary property of the signal by estimating the CAF of the received signal and detecting peaks in the resulting spectrum. The location of these peaks corresponds to the cyclic frequency of the signal, and their amplitude indicates the presence and strength of the signal.

Cyclostationary energy detection is a powerful technique that can improve the accuracy of spectrum sensing, particularly in low SNR environments. It can detect signals that are difficult to detect using other techniques, such as energy detection and matched filter detection. However, it requires a high computational complexity and is sensitive to the choice of the cyclic frequency estimation method.

In summary, cyclostationary energy is a property of a signal that exhibits periodic behavior in its statistical properties over time. Cyclostationary energy detection can be used in spectrum sensing techniques to improve the accuracy of signal detection in low SNR environments. It is a powerful technique, but it requires a high computational complexity and careful choice of the cyclic frequency estimation method.

5.2 MODULE DEVELOPMENT AND DESCRIPTION

1. Collect the datasets (audio signal)
2. Feature extraction
3. Applying algorithm
4. Remove the noises from audio.

1. Collect the data:

Audio signal is a representation of sound, typically using varying voltage levels for analog signals or a series of binary digits for digital signals. The frequency of the audio signal lies in the audio frequency range of approximately 20 to 20,000 Hz, which corresponds to the lower and upper limits of human hearing. Audio signals can be synthesized directly

or obtained from transducers such as microphones, instrument pickups, pickup units, and tape heads. Speakers or headphones convert electrical audio signals into sound.

Digital audio systems represent audio signals in various digital formats. An audio channel or audio track is an audio signal communication channel within a storage device or mixer used in operations such as multitrack recording.

2. Feature extraction:

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving information the original data set better results than applying machine learning directly to raw data.

3. Applying algorithm:

The Nonlocal mean (NLM) algorithm is an excellent spatial denoising algorithm [14–16]. The algorithm uses the characteristics of many similar blocks in the image and the average value of similar blocks to suppress image noise. At the beginning, Buades et al. [17] proposed a nonlocal average denoising algorithm based on the similarity (periodicity) of image features. Before this, the periodicity of the image was mainly used for texture synthesis. In recent years, there has been a lot of research and improvement on NLM algorithm denoising methods, which are mainly used for compressed sensing, feature extraction, pattern recognition, and image denoising.

The development of mathematical theory and image processing technology, good mathematical characteristics make its denoising advantages more obvious, and this algorithm has become one of the most well-known denoising algorithms. There are also many scholars who have improved the classic NLM algorithm, especially the steps of dividing the pixel area of the traditional algorithm, so that it can retain more image structure information [18]. For example, the patch-based NLM algorithm is used in the field of medical image recognition to improve the fidelity of the image [19].

4. Remove the noises from audio

Filtering image data is a standard process used in almost every image processing system. Filters are used for this purpose. They remove noise from images by preserving the details of the same. The choice of filter depends on the filter behaviour and type of data

5.3 DEVELOPMENT AND DEPLOYMENT

Machine Learning:

Machine learning is a field of computer science and artificial intelligence that focuses on the development of algorithms and statistical models that enable computers to learn from and make predictions or decisions based on data without being explicitly programmed. The goal of machine learning is to create systems that can automatically learn and improve from experience, without being explicitly programmed for every task.

There are three main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning involves training a model using labeled data, where the correct outputs are known, and the algorithm is trained to learn the relationship between inputs and outputs. For example, a supervised learning algorithm can be trained to classify images of animals as cats or dogs based on labeled data.

Unsupervised learning involves training a model using unlabeled data, where the algorithm must learn patterns and relationships in the data on its own. Clustering algorithms, which group similar data points together, are an example of unsupervised learning.

Reinforcement learning involves training a model to take actions in an environment to maximize a reward. The model learns through trial and error and is rewarded for making good decisions and penalized for making bad ones.

Machine learning has many applications in various fields, including finance, healthcare, transportation, and entertainment. Examples include fraud detection in financial transactions, predicting disease outbreaks, self-driving cars, and recommending movies or music based on user preferences.

Overall, machine learning is a rapidly evolving field with enormous potential for innovation and impact in various industries.

Support Vector Machine:

Support vector machines (SVMs) are a type of machine learning algorithm used for classification and regression analysis. They work by finding the best possible decision boundary that can separate different classes or groups in a dataset. In SVMs, the data points are represented as vectors in a multi-dimensional space, and the goal is to find a hyperplane that maximizes the margin between the different classes. The margin is the distance between the hyperplane and the closest data points from each class, and the SVM algorithm seeks to find the hyperplane that maximizes this margin.

SVMs are particularly useful for classification problems with a large number of features and where the classes are not easily separable. They can also be used for regression analysis, where the goal is to predict a continuous variable.

One advantage of SVMs is their ability to handle high-dimensional data with good accuracy, even when the number of features is greater than the number of observations. They are also less prone to overfitting than some other machine learning algorithms, and have a strong theoretical foundation.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.

5. Image Recognition – Distinguish the objects in an image.

The support vector machine is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. (Note that implementations differ in what they do at the edges of images, as explained below.) Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel.

However, SVMs can be sensitive to the choice of kernel function used to map the data to a higher-dimensional space, and can be computationally intensive for large datasets. They also may not perform well when the data is highly imbalanced, meaning that one class has many more examples than the other. Overall, SVMs are a powerful and widely-used tool in machine learning, with applications in fields such as image classification, natural language processing, and bioinformatics.

Another very important category is what we call image editing software. This group includes painting programs, graphic art packages and so on. They are often useful in conjunction with image processing software packages, in situations where direct immediate interaction with an image is the easiest way of achieving something. For instance, if a region of an image is to be masked out for subsequent image processing, it may be easiest to create the mask using an art package by directly drawing on top of the original image. The mask used in the description of the AND operator was created this way for instance.

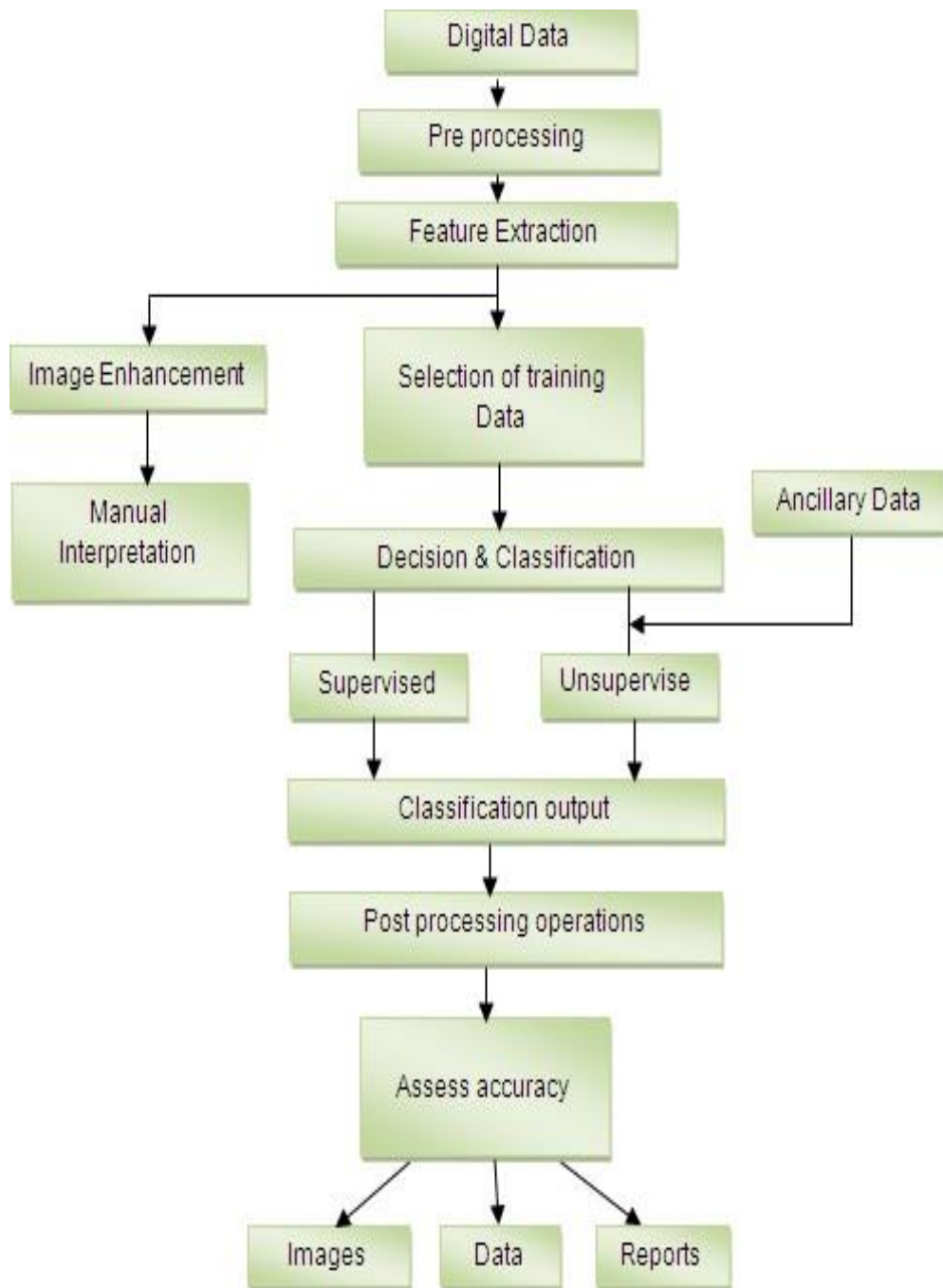


Fig 5.3.1 support vector machine block diagram

Convolution:

Convolution is a simple mathematical operation that is fundamental to many common image-processing operators. Convolution provides a way of 'multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values. In an image processing context, one of the input arrays is normally just a gray -level image. The second array is usually much smaller, is also two-dimensional (although it may be just a single pixel thick), and is known as the kernel. Figure 1 shows an example image and kernel that we will use to illustrate convolution.

I₁₁	I₁₂	I₁₃	I₁₄	I₁₅	I₁₆	I₁₇	I₁₈	I₁₉
I₂₁	I₂₂	I₂₃	I₂₄	I₂₅	I₂₆	I₂₇	I₂₈	I₂₉
I₃₁	I₃₂	I₃₃	I₃₄	I₃₅	I₃₆	I₃₇	I₃₈	I₃₉
I₄₁	I₄₂	I₄₃	I₄₄	I₄₅	I₄₆	I₄₇	I₄₈	I₄₉
I₅₁	I₅₂	I₅₃	I₅₄	I₅₅	I₅₆	I₅₇	I₅₈	I₅₉
I₆₁	I₆₂	I₆₃	I₆₄	I₆₅	I₆₆	I₆₇	I₆₈	I₆₉

K₁₁	K₁₂	K₁₃
K₂₁	K₂₂	K₂₃

Fig.5.3.2 Convolution

This Figure is an example small image (left) and kernel (right) to illustrate convolution. The labels within each grid square are used to identify each square. The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. (Note that implementations differ in what they do at the edges

of images, as explained below.) Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together.

So, in our example, the value of the bottom right pixel in the output image will be given by:

$$O_{57} = I_{57}K_{11} + I_{58}K_{12} + I_{59}K_{13} + I_{67}K_{21} + I_{68}K_{22} + I_{69}K_{23} \text{ ----- Eq (1)}$$

If the image has M rows and N columns, and the kernel has m rows and n columns, then the size of the output image will have M - m + 1 rows, and N - n + 1 columns.

Mathematically we can write the convolution as:

$$O(i, j) = \sum_{k=1}^m \sum_{l=1}^n I(i + k - 1, j + l - 1)K(k, l) \text{ ----- Eq (2)}$$

where i runs from 1 to M - m + 1 and j runs from 1 to N - n + 1.

Convolution can be used to implement many different operators, particularly spatial filters and feature detectors. Examples include Gaussian smoothing and the Sobel edge detector. The presence of the pixel grid makes several so-called distance metrics possible which often give different answers to each other for the distance between the same pair of points.

Distance Metrics:

It is often useful in image processing to be able to calculate the distance between two pixels in an image, but this is not as straightforward as it seems. The presence of the pixel grid makes several so-called distance metrics possible which often give different answers to each other for the distance between the same pair of points. We consider the three most important ones.

Euclidean Distance:

This is the familiar straight line distance that most people are familiar with. If the two pixels that we are considering have coordinates (x_1, y_1) and (x_2, y_2) , then the Euclidean distance is given by:

$$D_{Euclid} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \text{Eq (3)}$$

City Block Distance:

Also known as the Manhattan distance. This metric assumes that in going from one pixel to the other it is only possible to travel directly along pixel grid lines. Examples include Gaussian smoothing and the Sobel edge detector. Diagonal moves are not allowed. Therefore the 'city block' distance is given by:

$$D_{City} = |x_2 - x_1| + |y_2 - y_1| \text{Eq (4)}$$

Chessboard Distance:

This metric assumes that you can make moves on the pixel grid as if you were a King making moves in chess, *i.e.* a diagonal move counts the same as a horizontal move. The labels within each grid square are used to identify each square. This means that the metric is given by:

$$D_{Chess} = \max(|x_2 - x_1|, |y_2 - y_1|) \text{Eq (5)}$$

Note that the last two metrics are usually much faster to compute than the Euclidean metric and so are sometimes used where speed is critical but accuracy is not too important, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries.

Edge Detectors:

Edges are places in the image with strong intensity contrast. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. Representing an image by its edges has the further advantage that the amount of data is reduced significantly while retaining most of the image information.

Since edges correspond to strong illumination gradients, we can highlight them by calculating the derivatives of the image. This is illustrated for the one-dimensional case in Figure 1.

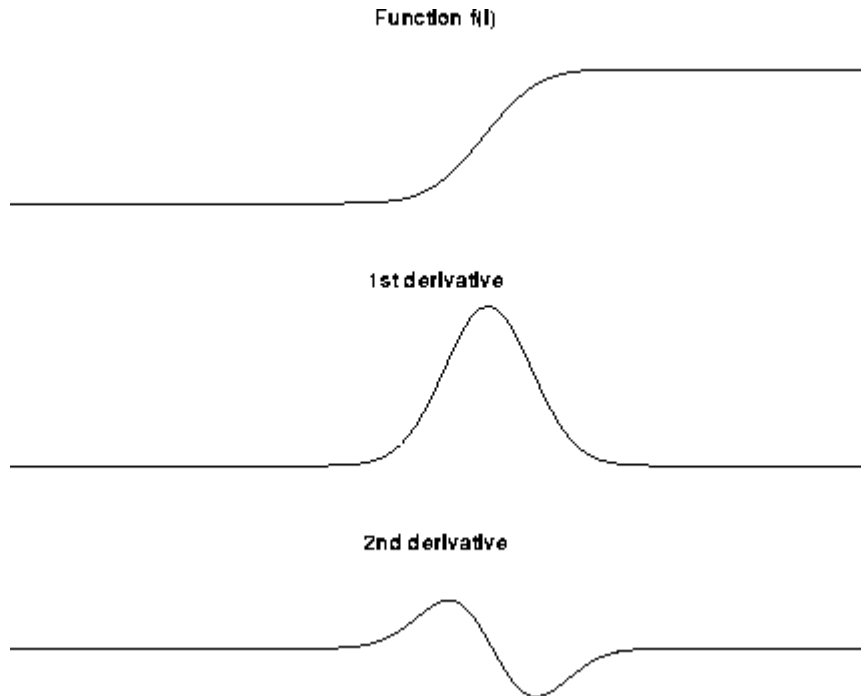


Fig 5.3.3 1st and 2nd derivative of an edge illustrated in one dimension.

We can see that the position of the edge can be estimated with the maximum of the 1st derivative or with the zero-crossing of the 2nd derivative. Therefore we want to find a

technique to calculate the derivative of a two-dimensional image. For a discrete one-dimensional function $f(i)$, the first derivative can be approximated by

$$\frac{df(i)}{di} = f(i+1) - f(i) \text{-----Eq (6)}$$

Calculating this formula is equivalent to convolving the function with $[-1 \ 1]$. Similarly the 2nd derivative can be estimated by convolving $f(i)$ with $[1 \ -2 \ 1]$.

Different edge detection kernels which are based on the above formula enable us to calculate either the 1st or the 2nd derivative of a two-dimensional image. There are two common approaches to estimate the 1st derivative in a two-dimensional image, Prewitt compass edge detection and gradient edge detection.

$$|G| = \sqrt{G_x^2 + G_y^2} \text{-----Eq (7)}$$

and is often approximated with

$$|G| = |G_x| + |G_y| \text{-----Eq (8)}$$

In many implementations, the gradient magnitude is the only output of a gradient edge detector, however the edge orientation might be calculated with

Image Editing Software:

There is a huge variety of software for manipulating images in various ways. Much of this software can be grouped under the heading image processing software, and the bulk of this reference is concerned with that group.

Another very important category is what we call image editing software. This group includes painting programs, graphic art packages and so on. They are often useful in conjunction with image processing software packages, in situations where direct immediate interaction with an image is the easiest way of achieving something. For instance, if a region of an image is to be masked out for subsequent image processing, it may be easiest to create the mask using an art package by directly drawing on top of the

original image. The mask used in the description of the AND operator was created this way for instance. Art packages also often allow the user to move sections of the images around and brighten or darken selected regions interactively. Few dedicated image processing packages offer the same flexibility and ease of use in this respect.

Mathematical Morphology:

The field of mathematical morphology contributes a wide range of operators to image processing, all based around a few simple mathematical concepts from set theory. The operators are particularly useful for the analysis of binary images and common usages include edge detection, noise removal, image enhancement and image segmentation.

The two most basic operations in mathematical morphology are erosion and dilation. Both of these operators take two pieces of data as input: an image to be eroded or dilated, and a structuring element (also known as a kernel). The two pieces of input data are each treated as representing sets of coordinates in a way that is slightly different for binary and grayscale images.

For a binary image, white pixels are normally taken to represent foreground regions, while black pixels denote background. (Note that in some implementations this convention is reversed, and so it is very important to set up input images with the correct polarity for the implementation being used). Then the set of coordinates corresponding to that image is simply the set of two-dimensional Euclidean coordinates of all the foreground pixels in the image, with an origin normally taken in one of the corners so that all coordinates have positive elements.

For a grayscale image, the intensity value is taken to represent height above a base plane, so that the grayscale image represents a surface in three-dimensional Euclidean space. Figure 1 shows such a surface. Then the set of coordinates associated with this image surface is simply the set of three-dimensional Euclidean coordinates of all the points within this surface and also all points below the surface, down to the base plane. Note that even when we are only considering points with integer coordinates, this is a lot of points, so usually algorithms are employed that do not need to consider all the points.

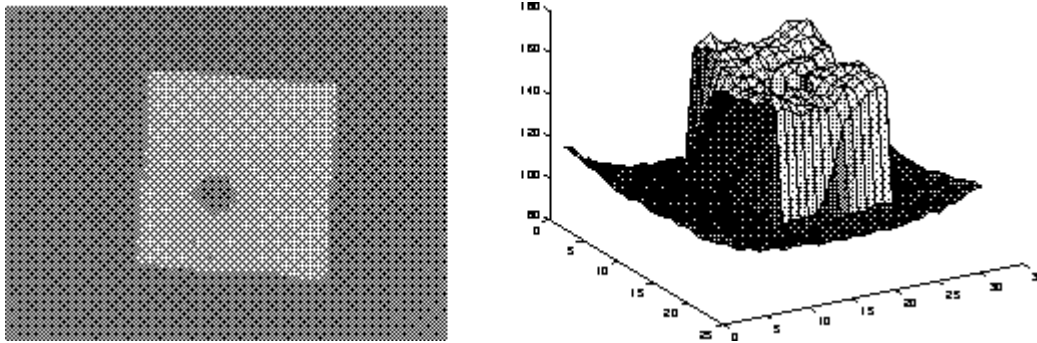


Fig 5.3.4 Simple gray level image and the corresponding surface in image space

The structuring element is already just a set of point coordinates (although it is often represented as a binary image). It differs from the input image coordinate set in that it is normally much smaller, and its coordinate origin is often not in a corner, so that some coordinate elements will have negative values. Note that in many implementations of morphological operators, the structuring element is assumed to be a particular shape (e.g. a 3×3 square) and so is hardwired into the algorithm.

Binary morphology can be seen as a special case of gray level morphology in which the input image has only two gray levels at values 0 and 1.

Erosion and dilation work (at least conceptually) by translating the structuring element to various points in the input image, and examining the intersection between the translated kernel coordinates and the input image coordinates. For instance, in the case of erosion, the output coordinate set consists of just those points to which the origin of the structuring element can be translated, while the element still remains entirely 'within' the input image.

Virtually all other mathematical morphology operators can be defined in terms of combinations of erosion and dilation along with set operators such as intersection and union. Some of the more important are opening, closing and skeletonization.

Multi-Spectral Images:

A multi-spectral image is a collection of several monochrome images of the same scene, each of them taken with a different sensor. Each image is referred to as a band. A well known multi-spectral (or multi-band image) is a RGB color image, consisting of a red, a green and a blue image, each of them taken with a sensor sensitive to a different wavelength. In image processing, multi-spectral images are most commonly used for Remote Sensing applications. Satellites usually take several images from frequency bands in the visual and non-visual range. Landsat 5, for example, produces 7 band images with the wavelength of the bands being between 450 and 1250 nm.

Special techniques exist to process multi-spectral images. For example, to classify a pixel as belonging to one particular region, its intensities in the different bands are said to form a feature vector describing its location in the k-dimensional feature space. The simplest way to define a class is to choose an upper and lower threshold for each band, thus producing a k-dimensional 'hyper-cube' in the feature space. Only if the feature vector of a pixel points to a location within this cube, is the pixel classified as belonging to this class. A more sophisticated classification method is described in the corresponding worksheet.

The disadvantage of multi-spectral images is that, since we have to process additional data, the required computation time and memory increase significantly. However, since the speed of the hardware will increase and the costs for memory will decrease in the future, it can be expected that multi-spectral images will become more important in many fields of computer vision.

Non-Linear Filtering:

Suppose that an image processing operator F acting on two input images A and B produces output images C and D respectively. If the operator F is linear, then

$$F(a \times A + b \times B) = a \times C + b \times D \text{ -----Eq (9)}$$

where a and b are constants. In practice, this means that each pixel in the output of a linear operator is the weighted sum of a set of pixels in the input image.

By contrast, non-linear operators are all the other operators. For example, the threshold operator is non-linear, because individually, corresponding pixels in the two images A and B may be below the threshold, whereas the pixel obtained by adding A and B may be above threshold. Similarly, an absolute value operation is non-linear:

$$|-1 + 1| \neq |-1| + |1| \text{ -----Eq (10)}$$

as is the exponential operator:

$$\exp(1 + 1) \neq \exp(1) + \exp(1) \text{ -----Eq (11)}$$

Pixels:

In order for any digital computer processing to be carried out on an image, it must first be stored within the computer in a suitable form that can be manipulated by a computer program. The most practical way of doing this is to divide the image up into a collection of discrete (and usually small) cells, which are known as pixels. Most commonly, the image is divided up into a rectangular grid of pixels, so that each pixel is itself a small rectangle. Once this has been done, each pixel is given a pixel value that represents the color of that pixel. It is assumed that the whole pixel is the same color, and so any color variation that did exist within the area of the pixel before the image was discretized is lost. However, if the area of each pixel is very small, then the discrete nature of the image is often not visible to the human eye.

Other pixel shapes and formations can be used, most notably the hexagonal grid, in which each pixel is a small hexagon. This has some advantages in image processing, including the fact that pixel connectivity is less ambiguously defined than with a square grid, but hexagonal grids are not widely used. Part of the reason is that many image capture systems (e.g. most CCD cameras and scanners) intrinsically discretize the captured image into a rectangular grid in the first instance.

Pixel Connectivity:

The notation of pixel connectivity describes a relation between two or more pixels. For two pixels to be connected they have to fulfill certain conditions on the pixel brightness and spatial adjacency.

First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values V . For a grayscale image, V might be any range of graylevels, e.g. $V=\{22,23,...40\}$, for a binary image we simple have $V=\{1\}$.

To formulate the adjacency criterion for connectivity, we first introduce the notation of neighborhood. For a pixel p with the coordinates (x,y) the set of pixels given by:

$$N_4(p) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\} \text{ -----Eq (12)}$$

is called its 4-neighbors. Its 8-neighbors are defined as

$$N_8(p) = N_4 \cup \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\} \text{ -----Eq (13)}$$

From this we can infer the definition for 4- and 8-connectivity:

Two pixels p and q , both having values from a set V are 4-connected if q is from the set $N_4(p)$ and 8-connected if q is from $N_8(p)$.

General connectivity can either be based on 4- or 8-connectivity; for the following discussion we use 4-connectivity.

A pixel p is connected to a pixel q if p is 4-connected to q or if p is 4-connected to a third pixel which itself is connected to q . Or, in other words, two pixels q and p are connected if there is a path from p and q on which each pixel is 4-connected to the next one.

A set of pixels in an image which are all connected to each other is called a connected component. Finding all connected components in an image and marking each of them with a distinctive label is called connected component labeling.

An example of a binary image with two connected components which are based on 4-connectivity can be seen in Figure 1. If the connectivity were based on 8-neighbors, the two connected components would merge into one.

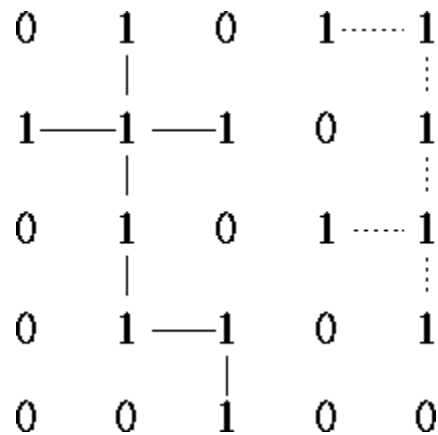


Fig 5.3.5 Two connected components based on 4-connectivity.

Primary Colors:

It is a useful fact that the huge variety of colors that can be perceived by humans can all be produced simply by adding together appropriate amounts of red, blue and green colors. These colors are known as the primary colors. Thus in most image processing applications, colors are represented by specifying separate intensity values for red, green and blue components. This representation is commonly referred to as RGB.

The primary color phenomenon results from the fact that humans have three different sorts of color receptors in their retinas which are each most sensitive to different visible light wavelengths.

The primary colors used in painting (red, yellow and blue) are different. When paints are mixed, the 'addition' of a new color paint actually subtracts wavelengths from the reflected visible light.

CHAPTER 6

SOFTWARE DESCRIPTION

MATLAB:

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

Matlab is a data analysis and visualization tool which has been designed with powerful support for matrices and matrix operations. As well as this, Matlab has excellent graphics capabilities, and its own powerful programming language. One of the reasons that Matlab has become such an important tool is through the use of sets of Matlab programs designed to support a particular task. These sets of programs are called toolboxes, and the particular toolbox of interest to us is the image processing toolbox. Rather than give a description of all of Matlab's capabilities, we shall restrict ourselves to just those aspects concerned with handling of images. We shall introduce functions, commands and techniques as required. A Matlab function is a keyword which accepts various parameters, and produces some sort of output: for example a matrix, a string, a graph. Examples of such functions are `sin`, `imread`, `imclose`. There are many functions in Matlab, and as we shall see, it is very easy (and sometimes necessary) to write our own.

MATLAB And Images

- The help in MATLAB is very good, use it!
- An image in MATLAB is treated as a matrix
- Every pixel is a matrix element
- All the operators in MATLAB defined on matrices can be used on images: `+`, `-`, `*`, `/`, `^`, `sqrt`, `sin`, `cos` etc.
- MATLAB can import/export several image formats

Image Type Conversion

- RGB Image to Intensity Image (rgb2gray)
- Indexed Image to RGB Image (ind2rgb)
- Intensity Image to RGB Image (gray2ind, ind2rgb)

Key Features

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces

PYTHON:

Python is a high-level, general-purpose programming language that was first released in 1991. It is known for its simplicity, readability, and ease of use, which makes it an ideal language for beginners to learn. Python code is typically written in a clear and concise manner, making it easy to read and understand.

Python supports multiple programming paradigms, including procedural, functional, and object-oriented programming. It has a large standard library, which provides access to a wide range of modules and functions for tasks such as file I/O, networking, and database access. Additionally, Python has a vast ecosystem of third-party libraries and tools, which makes it a popular language for a variety of applications, including web development, scientific computing, data analysis, and machine learning.

One of the key features of Python is its dynamic type system, which allows for flexible and rapid development. Python also supports automatic memory management through a process called garbage collection, which frees developers from having to manually manage memory allocation and deallocation.

Overall, Python is a powerful and versatile programming language that has gained popularity for its ease of use, readability, and vast ecosystem of libraries and tools. Python also supports automatic memory management through a process called garbage collection, which frees developers from having to manually manage memory allocation and deallocation.

1. **Easy to Learn:** Python has a simple and intuitive syntax that is easy to learn and read, making it a popular choice for beginners.
2. **Interpreted:** Python is an interpreted language, meaning that code is executed line-by-line, making it easy to test and debug code.
3. **Object-Oriented:** Python supports object-oriented programming (OOP), allowing developers to create reusable and modular code.
4. **High-Level Language:** Python is a high-level language, which means that it has built-in abstractions that allow developers to write code in a way that is closer to natural language.
5. **Cross-Platform:** Python is a cross-platform language, meaning that code written on one platform can be run on another without modification.
6. **Large Standard Library:** Python comes with a large standard library that provides pre-written code for many common tasks, such as networking, file I/O, and regular expressions.
7. **Third-Party Libraries:** Python has a large and active community of developers who create third-party libraries that extend its functionality, making it suitable for a wide range of applications.
8. **Dynamically Typed:** Python is a dynamically typed language, meaning that the type of a variable is determined at runtime rather than at compile time, allowing for greater flexibility and faster development times.

CHAPTER 7

CONCLUSION

CONCLUSION:

The first code implements cooperative spectrum sensing in a cognitive radio network. The simulation calculates the probability of missed detection (P_{md}) versus the probability of false alarm (P_{fa}) for an energy detector under AWGN (Additive White Gaussian Noise) channel conditions. The simulation considers a network with C_{rs} (Cognitive Radio Users), where each user implements local spectrum sensing to detect a primary user's signal, and the individual results are combined to make the final decision. The simulation implements an AND rule for combining individual decisions. The results of the simulation are compared with the theoretical analysis of P_{md} versus P_{fa} .

The second code optimizes the spectrum sensing of a cognitive radio network. The optimization considers the detection probability and the probability of false alarm for an energy detector under AWGN channel conditions. The code computes the probability of missed detection (P_m) versus the probability of false alarm (P_f) for different values of the sensing time. The code then computes the average probability of missed detection (P_{mi}) and the average probability of detection (P_{dc}) for multiple runs, considering the optimization of the sensing time to minimize the total error ($P_m + P_f$). The optimization is performed using a for loop that varies the sensing time and computes the total error for different values of the sensing time. The total error is then averaged over multiple runs to obtain the average P_{mi} and P_{dc} for each sensing time. The total error is then averaged over multiple runs to obtain the average P_{mi} and P_{dc} for each sensing time. Finally, the results of the simulation are plotted to show the variation of P_{mi} and P_{dc} with sensing time.

In summary, both codes implement spectrum sensing for cognitive radio networks. The first code uses a cooperative approach where the decisions of multiple users are combined to make the final decision. The second code optimizes the sensing time to minimize the total error ($P_m + P_f$).

Complementary ROC of Cooperative Sensing with AND Rule under AWGN:

The graph shows the complementary receiver operating characteristic (ROC) curves for cooperative sensing with the AND rule under additive white Gaussian noise (AWGN) channel. The x-axis represents the probability of false alarm (P_{fa}), while the y-axis represents the probability of missed detection (P_{md}).

The red asterisk (*), green asterisk (*), and black circle (o) lines represent the simulated, theoretical, and practical probabilities of missed detection, respectively. The simulation results were obtained by running a Monte Carlo simulation to estimate the P_{md} for different values of P_{fa} . The theoretical results were computed using mathematical equations, while the practical results were obtained using a threshold value derived from the signal-to-noise ratio (SNR) and the time-bandwidth product (η).

The graph shows that as the P_{fa} increases, the P_{md} also increases. The theoretical curve is the closest to the simulated curve, while the practical curve is slightly below the other two curves, which implies that the practical threshold value used in the simulation is conservative.

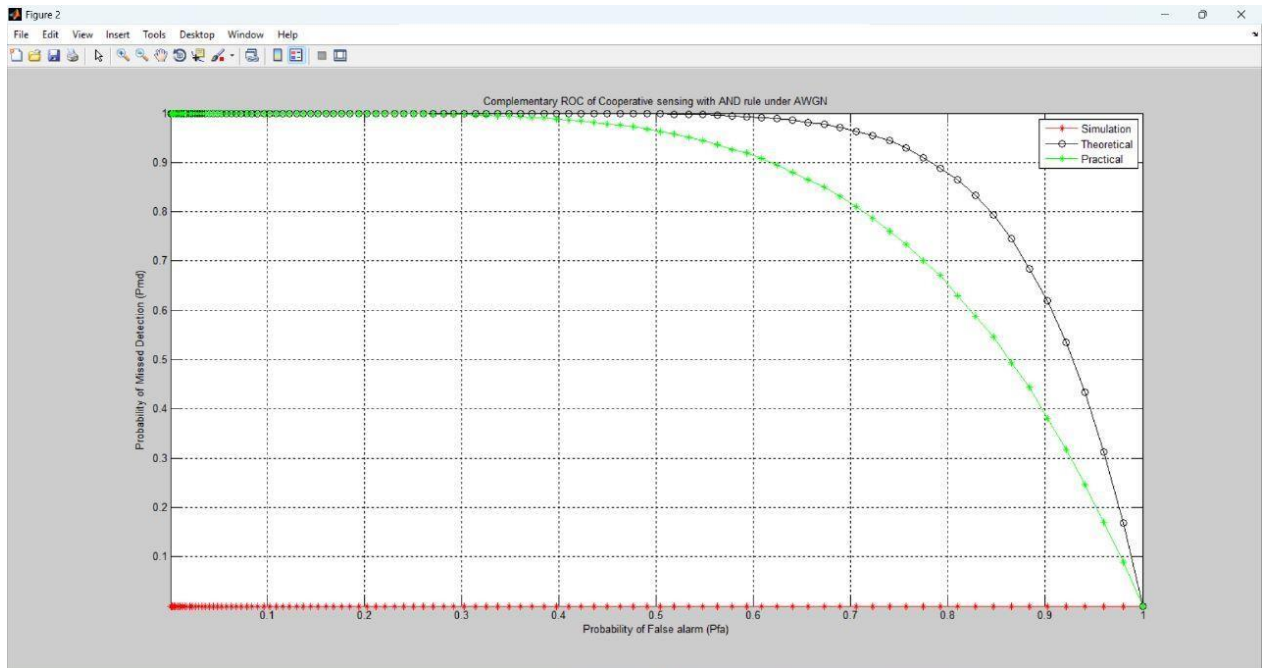


Fig 6.1 Complementary ROC of Cooperative Sensing with AND Rule under AWGN

Comparison of Total Error Rates for Spectrum Sensing in Cognitive Radio Network using Energy Detection and Matched Filter Detection:

This program compares the total error rates for spectrum sensing in a cognitive radio network using energy detection and matched filter detection. The graph shows the total error rates for both methods for different values of n (the number of samples used for energy detection) and for a range of thresholds. The lines represent the total error rate for matched filter detection for $n=1$ to $n=10$, while the red asterisks represent the total error rate for energy detection with $n=5$.

The total error rate is a measure of the performance of spectrum sensing in cognitive radio networks, which is the ability of a radio to detect whether a frequency band is vacant or occupied. The total error rate is the sum of the probability of false alarm (P_f) and the probability of missed detection (P_m). The program computes the total error rate for different values of the threshold, which is used to distinguish between the signal and noise. The program also computes the probability of detection (P_d), which is the probability of correctly detecting the signal, and the probability of false alarm (P_f), which is the probability of detecting the signal when there is only noise.

The graph shows that for a given threshold, the total error rate is smaller for matched filter detection than for energy detection. This is because matched filter detection is more reliable in detecting signals in a noisy environment. The graph also shows that increasing the number of samples used for energy detection (n) leads to a decrease in the total error rate. This is because using more samples reduces the effect of noise on the detection of the signal.

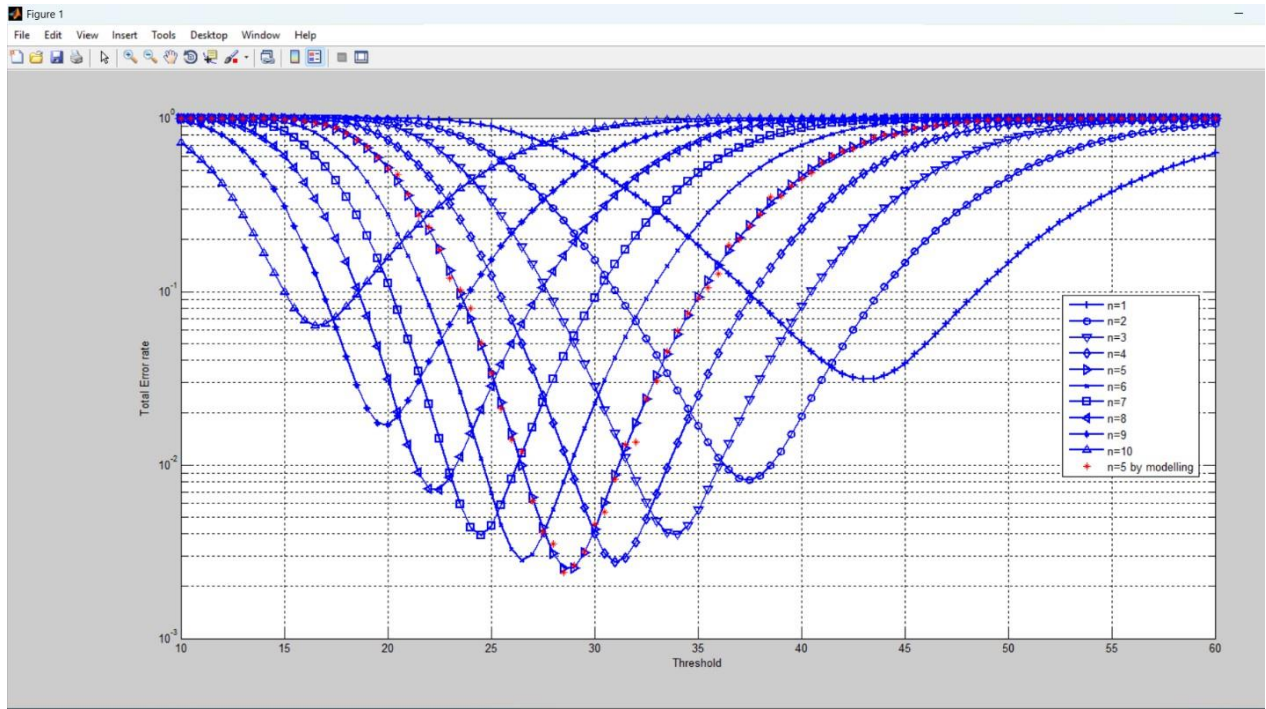


Fig 6.2 Comparison of Total Error Rates using Energy Detection and Matched Filter Detection

REFERENCES

1. Ahmad, W.S.H.M.W.; Radzi, N.A.M.; Samidi, F.; Ismail, A.; Abdullah, F.; Jamaludin, M.Z.; Zakaria, M. 5G technology: Towards dynamic spectrum sharing using cognitive radio networks. *IEEE Access* 2020, 8, 14460–14488.
2. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, *IEEE Commun.Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
3. Amjad, M.; Akhtar, F.; Rehmani, M.H.; Reisslein, M.; Umer, T. Full-Duplex Communication in Cognitive Radio Networks: A Survey. *IEEE Commun. Surv. Tutor.* 2017, 19, 2158–2191
4. Arjoun, Y.; Kaabouch, N. A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions. *Sensors* 2019, 19, 126. [CrossRef]
5. Bhutta, M. N., & Zaidi, S. A. R. (2019). A survey on spectrum sensing for cognitive radio networks using machine learning techniques. *International Journal of Communication Systems*, 32(3), e3963.
6. Cheng, Y., Chen, X., Chen, Q., & Huang, H. (2020). Enhanced spectrum sensing in cognitive radio networks: Recent advances and future challenges. *IEEE Communications Magazine*, 58(10), 28-33.
7. Chia-Yen Shih; Lee, W.C.; Kuo, C.C.J.; Wei-Cheng Wang; Chih-Yang Chang; Jyh-Horng Wen, "A Lightweight Convolutional Neural Network for Cooperative Spectrum Sensing in Cognitive Radio Networks," in *IEEE Access*, vol. 7, pp. 5467-5478, 2019.
8. D'Amico, A.; Rondón-Vallejo, P.; Vidal, J. IoT meets 5G in the era of tactile internet: Applications and challenges. *Sensors* 2019, 19, 259. [CrossRef]
9. F. A. P. de Figueiredo, D. Stojadinovic, P. Maddala, R. Mennes, I. Jabandžić, X. J. Jiao, and I. Moerman, 'Scatter Phy: A physical layer for the DARPA spectrum collaboration challenge, in *Proc. 2019 IEEE Int. Symp. Dynamic Spectrum Access Networks (DySPAN)*, Newark, NJ, USA, 2019, pp. 1–6

10. Fujitsu Laboratories Ltd, "Fujitsu Develops Low-power Consumption LSI for Deep Learning Inferencing," in Fujitsu Laboratories Ltd, 2019.
11. Gupta, M.S.; Kumar, K. Progression on spectrum sensing for cognitive radio networks: A survey, classification, challenges and future research issues. *J. Netw. Comput. Appl.* 2019, 143, 47–76.
12. Gao, X., Zhang, J., Zhou, L., & Li, B. (2020). Hybrid spectrum sensing in cognitive radio networks: Recent advances and future challenges. *IEEE Communications Magazine*, 58(10), 22-27.
13. Habib, M. A., Iqbal, S., & Anpalagan, A. (2020). A survey on spectrum sensing techniques in cognitive radio networks: Advances and challenges. *IEEE Communications Surveys & Tutorials*, 22(3), 1603-1633.
14. He, J.; Zhang, S.; Zhang, J.; Jin, L.; Hu, H. A cooperative spectrum sensing algorithm based on deep belief networks for cognitive radio networks. *J. Netw. Comput. Appl.* 2019, 126, 106–114. [CrossRef]
15. Huang, Q., Zhang, X., & Wang, C. X. (2018). Deep learning for spectrum sensing: A review. *IEEE Communications Magazine*, 56(9), 110-117.

APPENDIX

SOURCE CODE

Cooperative Spectrum Sensing for Cognitive Radio Networks Under AWGN Channel: A Comparative Study between Theoretical, Practical and Simulation Results

```
clear all;
close all;
clc

u=1000;%time bandwidth factor
N=2*u;%samples
a=2;%path loss exponent
C=2;%constant losses
Crs=10; %Number of cognitive radio users
PdAnd=0;
%- ---- -Pfa------%
Pf=0.01:0.01:1;
Pfa=Pf.^2;
%-----signal ---- %
t=1:N;
s1 = cos(pi*t);
stem(t,s1)
s1power=var(s1);

%- ---- SNR ------%
% Snrdb=-15:1:15;
Snrdb=15;
Snreal=power(10,Snrdb/10);%Linear Snr
```

```

% while Snrdb<15%
lamda=ones(1,100);
for i=1:length(Pfa)
lamda(i)=gammaincinv(1-Pfa(i),u)*2; %theshold
% lamdadB=10*log10(lamda);
%-----Local spectrum sensing ----- %
d=ones(1,10);
for j=1:CrS %for each node
detect=0;
d(j)=7+1.1*rand(); %random distanse
PL=C*(d(j)^-a); %path loss

for sim=1:10%Monte Carlo Simulation for 100 noise realisation

%----- AWGN channel-----%
noise = randn(1,N); %Noise production with zero mean and s^2 var
noise_power = mean(noise.^2); %noise average power
amp = sqrt(noise.^2*Snreal);
s1=amp.*s1./abs(s1);
% SNRdB_Sample=10*log10(s1.^2./(noise.^2));
Rec_signal=s1+noise;%received signal
localSNR=ones(1,10);
localSNR(j)=mean(abs(s1).^2)*PL/noise_power;%local snr
pdth=cell(10,100);
pdth{1,100}=1:100;
pdth{10,10}='string';
Pdth(j,i)=marcumq(sqrt(2*localSNR(j)),sqrt(lamda(i)),u);%Pd for j node

%Computation of Test statistic for energy detection
Sum=abs(Rec_signal).^2*PL;

```

```

Test=ones(1,10);
Test(j,sim)=sum(Sum)
if (Test(j,sim)>lamda(i))
detect=detect+1;
end
end %END Monte Carlo
Pdsim=ones(1,10);
Pdsim(j)=detect/sim; %Pd of simulation for the j-th CRuser
end
PdAND=ones(1,100);
PdAND(i)=prod(Pdsim);
PdOR=ones(1,100);
PdOR(i)=1-prod(1-Pdsim);

end
PdAND5=(Pdth(5,:)).^5;
Pmd5=1-PdAND5;
PdANDth=(Pdth(Crs,:)).^Crs;
PmdANDth=1-PdANDth; %Probability of miss detection
Pmdsim=1-PdAND;
figure(2);
plot(Pfa,Pmdsim,'r-*',Pfa,PmdANDth,'k-o',Pfa,Pmd5,'g-*');
title('Complementary ROC of Cooperative sensing with AND rule under AWGN');
grid on
axis([0.0001,1,0.0001,1]);
xlabel('Probability of False alarm (Pfa)');
ylabel('Probability of Missed Detection (Pmd)');
legend('Simulation','Theoretical','Practical');

```

The code is an implementation of cooperative spectrum sensing for cognitive radio networks using the energy detection technique. In cognitive radio networks, the

secondary users (CR users) can utilize the unused portions of the radio spectrum to transmit data without interfering with the primary users' communication. However, before transmitting on any frequency band, it is crucial to sense whether that frequency is occupied by the primary user or not.

The energy detection technique used in this code senses the energy level of the received signal at the CR user's receiver. The energy detector is based on a threshold test that compares the energy level of the received signal to a predefined threshold. If the energy level of the received signal is above the threshold, the CR user decides that the primary user is occupying the frequency band.

The code is written in MATLAB and starts by defining various parameters such as the time bandwidth factor (α), the number of samples (N), the path loss exponent (α), constant losses (C), and the number of cognitive radio users (C_r s). The code then computes the probability of false alarm (P_{fa}) using the formula $P_{fa} = P_f^2$ for various values of P_f .

Next, the code generates a signal, s_1 , which is a cosine wave. The power of s_1 is also calculated. The code then computes the signal-to-noise ratio (SNR) for various values of SNRdB. The AWGN channel is modeled by adding noise to the received signal. The power of the noise is calculated, and the received signal's amplitude is adjusted to achieve the desired SNR.

The code then performs local spectrum sensing for each CR user. For each CR user, a path loss is computed based on the distance between the CR user and the primary user. The energy level of the received signal is computed, and the local SNR is calculated.

The code then computes the probability of detection (P_d) for each CR user using the threshold test. The P_d is calculated using both Monte Carlo simulation and theoretical analysis. The theoretical P_d is calculated using the Marcum Q-function.

Finally, the code computes the P_d for the cooperative sensing using the AND rule, where the CR users transmit their sensing results to a fusion center that combines them using the AND rule. The code plots the complementary receiver operating characteristic (ROC)

curves for the cooperative sensing using the AND rule for various values of P_{fa} . The ROC curves show the trade-off between the probability of false alarm and the probability of missed detection.

In conclusion, the code demonstrates the energy detection-based cooperative spectrum sensing technique for cognitive radio networks. The code shows that the P_d improves as the number of CR users increases, which enhances the system's reliability. The complementary ROC curves show the performance of the cooperative sensing with the AND rule and provide insights into the system's performance trade-offs. Overall, this code provides a useful tool for the analysis and design of cognitive radio networks.

Optimization of Spectrum Sensing in Cognitive Radio Networks using Matched Filter and Energy Detection

```
clc;
close all;
clear all;
%This program is for optimization of spectrum sensing in
%Cognitive radio network.
N=20;
j=1;
tt=[];
err2=[];
Pmi=[];
Pdc=[];
error=[];
err1=[];
K=10;
snr=10;
Qd=0;
Qf=0;
tt=10:0.5:60;
vec=['-+', '-o', '-v', '-d', '->', '-x', '-s', '-<', '-*', '-^'];
```

```

for n=1:1:10

s=ones(1,N);
w=randn(1,N);

u=N/2;           %Time-delay bandwidth product

for t=10:0.5:60

Qd=0;
Qf=0;
SNR=10^(snr/10); %for linear scale

a=sqrt(2*SNR);
b=sqrt(t);

Pd = marcumq(a,b,u);      % AVG. PROB OF DETECTION(computes the generalized
Marcum Q)

Pf = gammainc((t/2),u,'upper');% AVG. PROB OF FALSE ALARM(compute incomplete
gamma function)

Pm=1-Pd;           %AVG. PROB OF MISSED DETECTION OVER AWGN
for l=n:1:K
Qd=Qd+(factorial(K)*(Pd^l)*((1-Pd)^(K-l))/(factorial(l)*factorial(K-l)));
Qf=Qf+(factorial(K)*(Pf^l)*((1-Pf)^(K-l))/(factorial(l)*factorial(K-l)));
end

Qm=1-Qd;

```

```

err=Qf+Qm;
err1=[err1 err];
end

end

l=1;
i=1;
for j=1:1:10
semilogy(tt,err1(i:i+100),vec(l:l+1),'LineWidth',1.5)
i=i+101;
l=l+2;
hold on;
end

grid on;
ylabel('Total Error rate');
xlabel('Threshold');

%- ----- Energy Detection -----
n=5;
rel=10000;
tt1=10:0.5:60;
er1=[];
for t=10:0.5:60
Pdc=0;
Pfc=0;
Qd=0;
Qf=0;
Qm=0;

for i=1:1:rel
SNR=10;

```

```

snr=10^(SNR/10);

s=ones(1,N);
w=randn(1,N);
vari=var(w);           %variance of noise

Es=sum(s.^2);

N02=(Es)/(2*snr);

x1=s+w;
x2=w;
W=1;                   %Time-delay bandwidth product

E0=(sum(x2.^2))/((W*N02));

E1=(sum(x1.^2))/((W*N02));

if E1>t
    Pdc=Pdc+1;
else
end
if E0>t
    Pfc=Pfc+1;
else
end
end
Pd=Pdc/rel;
Pf=Pfc/rel;

```

```

for l=n:1:K
Qd=Qd+(factorial(K)*(Pd^l)*((1-Pd)^(K-l))/(factorial(l)*factorial(K-l)));
Qf=Qf+(factorial(K)*(Pf^l)*((1-Pf)^(K-l))/(factorial(l)*factorial(K-l)));
end
Qm=1-Qd;
er=Qf+Qm;
er1=[er1 er];
end
hold on;
semilogy(tt1,er1,'*r')
grid on;

ylabel('Total Error rate');
xlabel('Threshold');
legend('n=1','n=2','n=3','n=4','n=5','n=6','n=7','n=8','n=9','n=10','n=5 by modelling');

```

The code is written in MATLAB and is used to optimize the spectrum sensing in cognitive radio networks. In cognitive radio networks, the spectrum is a valuable resource that needs to be managed efficiently. Spectrum sensing is an essential part of cognitive radio networks that enables the identification of available frequency bands that can be utilized by the cognitive radio.

The code uses two different methods for spectrum sensing, matched filter detection, and energy detection. Matched filter detection is a technique used to detect a known signal in noise, and it involves cross-correlating the received signal with a known signal. Energy detection, on the other hand, is a technique that involves computing the energy of the received signal and comparing it to a threshold.

The code computes the probability of detection (P_d), probability of false alarm (P_f), and probability of missed detection (P_m) using the Marcum Q function and incomplete gamma function. These probabilities are then used to compute the total error rate for different thresholds. The program also includes a loop that runs the optimization for different values of n (number of samples), and plots the total error rate versus threshold for each value of n using different line styles.

In the energy detection section, the code generates random noise and signal samples, computes the energy of the received signal, and determines the probability of detection and false alarm. The total error rate is then computed and plotted versus threshold.

In conclusion, the project aims to optimize the spectrum sensing in cognitive radio networks using two different methods, matched filter detection, and energy detection. The project provides insights into the performance of these methods and their effectiveness in identifying available frequency bands in a cognitive radio network. The code provided serves as a useful tool for simulating the performance of these methods and for optimizing the threshold values.