University ABBES LAGHROUR Khenchela
Faculty of Sciences and Technology
Department of Industrial Engineering
جامـــعة عبـــاس لـــــغرور خـنـشـلة
كليـــة العلـــوم والتكنـــولـــوجيـا
قسم الهنـدسة الصنـاعية

Serial Number :.....

Master's Thesis

For the obtaining of the Master's degree

Field: Telecommunications
Specialty: Telecommunications Systems

ENTITLED

Realization of a Smart Security and Monitoring System Using Artificial Intelligence Methods and Wireless Transmission Technologies

Submitted by **:** *Chaima ATTALAH*

Amel HAMIDANE

Defended on: 19/06/2023 ***Before the jury composed of:***

Pr. BEDRA Sami	***President***	***University Abbes Laghrour-Khenchela***
Dr. DOUAK Fouzi	***Supervisor***	***University Abbes Laghrour-Khenchela***
Dr. KHEZZAR Zaki Aissam	***Examiner***	***University Abbes Laghrour-Khenchela***

Promotion 2022/2023

Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامـــعة عبـــاس لـــغرور خـنـشـلة
كليـــة العلـــوم والتكنـــولـــوجيـا
قسم الهنـدسة الصنـاعية

N° Série :.....

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière: Télécommunications
Spécialité: Systèmes des Télécommunications

THEME

Réalisation d'un Système Intelligent de Sécurité et de Surveillance en Utilisant des Méthodes d'Intelligence Artificielle et des Technologies de Transmission Sans Fil

Réalisé par : *ATTALAH Chaima*
HAMIDANE Amel

Soutenu le : 19/06/2023 ***Devant le jury composé de:***

Pr. BEDRA Sami	***Président***	***Université Abbes Laghrour-Khenchela***
Dr. DOUAK Fouzi	***Encadreur***	***Université Abbes Laghrour-Khenchela***
Dr. KHEZZAR Zaki Aissam	***Examinateur***	***Université Abbes Laghrour-Khenchela***

Promotion 2022/2023

To my beloved parents for their unwavering love, support, and encouragement throughout my academic journey. Their sacrifices, guidance, and prayers have been the driving force behind my success.

To whom I have shared all my life, my two wonderful sisters, Achouak and Bouchra, for their constant support and motivation.

To all those who have supported and encouraged me throughout this journey.

To everyone who contributed to spreading knowledge freely.

Amel

I dedicate this dissertation to my dear parents, Abdelkarim and Aicha Nechneche, as they were the main support and source of encouragement and motivation throughout the study period. Thank you for everything. Accomplishing this would hopefully make you proud of me, as much as I am proud of having you as my parents. I also extend my greetings and sincere thanks to my sweet sister, Nada, who was always there to share in the joys and challenges.

I would like to sincerely thank our supervisor, Mr. Douak for your guidance, support, and patience throughout this study, thank you for the comments and questions you shared that is beneficial in the completion of this study.

To my friends who have supported me throughout the process. I will always appreciate all they have done.

Chaima

Acknowledgments

This work is the culmination of hard work and a lot of sacrifice; our thanks go first to the Creator of the universe who endowed us with intelligence, and kept us healthy to carry out this year of study.

We would like to express our gratitude to the supervisor of this thesis, Dr. Fouzi DOUAK, for his patience, his availability and above all his judicious advice, which helped to fuel our reflection.

We are also grateful to Dr. Zaki Aissam KHEZZAR for accepting to be a member of our jury and devoting their precious time to review our work. We also want to thank Professor Sami BEDRA for agreeing to preside over this jury.

We would like to express our deepest and sincere gratitude to all our teachers who allow us to acquire more knowledge during our journey.

Finally, we wish you a good reading.

Abstract

As security concerns grow, there is a greater demand for effective surveillance systems capable of providing real-time monitoring and identification of security threats. Traditional surveillance technologies are frequently restricted in capability and unreliable in critical situations. This work proposes the construction of a smart security system that uses artificial intelligence (AI) approaches and wireless transmission technologies to address this issue. To improve its detection and recognition capabilities, the suggested system employs a variety of AI algorithms, including machine learning and computer vision. Furthermore, the system employs wireless communication technologies such as Radio Frequency and GSM to provide remote monitoring and control. The system is intended to be scalable, cost-effective, and simple to implement, making it appropriate for usage in a variety of contexts such as homes, offices, and public spaces. The proposed security system provides a dependable and effective solution for assuring safety and security in a variety of locations by combining AI and wireless transmission technologies.

Key words: NVIDIA Jetson Nano, Artificial Intelligence, Deep Learning, Arduino, GSM, and Radio Frequency.

Résumé

À mesure que les problèmes de sécurité augmentent, il existe une demande croissante pour des systèmes de surveillance efficaces capables de fournir une surveillance et une identification en temps réel des menaces de sécurité. Les technologies de surveillance traditionnelles ont souvent des capacités limitées et ne sont pas fiables dans les situations critiques. Ce mémoire propose la construction d'un système de sécurité intelligent qui utilise des approches d'intelligence artificielle (AI) et des technologies de transmission sans fil pour résoudre ce problème. Pour améliorer ses capacités de détection et de reconnaissance, le système suggéré utilise une variété d'algorithmes d'AI, y compris l'apprentissage automatique et la vision par ordinateur. De plus, le système utilise des technologies de communication sans fil telles que la radiofréquence et le GSM pour assurer la surveillance à distance. Le système est conçu pour être évolutif, rentable et simple à mettre en œuvre, ce qui le rend approprié pour une utilisation dans une variété de contextes tels que les maisons, les bureaux et les espaces publics. Le système de sécurité proposé fournit une solution fiable et efficace pour assurer la sûreté et la sécurité dans une variété d'endroits en combinant l'AI et les technologies de transmission sans fil.

Mots clés : NVIDIA Jetson Nano, Intelligence Artificielle, Apprentissage Profond, Arduino, GSM, et Radio Fréquence.

ملخص

مع تزايد المخاوف الأمنية، هناك طلب أكبر على أنظمة المراقبة الفعالة القادرة على توفير المراقبة في الوقت الحقيقي وتحديد التهديدات الأمنية. غالبًا ما تكون تقنيات المراقبة التقليدية مقيدة من حيث القدرة ولا يمكن الاعتماد عليها في المواقف الحرجة. تقترح هذه المذكرة إنشاء نظام أمان ذكي يستخدم أساليب الذكاء الاصطناعي (AI) وتقنيات الإرسال اللاسلكي لمعالجة هذه المشكلة. لتحسين قدرات الكشف والتعرف، يستخدم النظام المقترح مجموعة متنوعة من خوارزميات الذكاء الاصطناعي، بما في ذلك التعلم الآلي ورؤية الكمبيوتر. علاوة على ذلك، يستخدم النظام تقنيات الاتصال اللاسلكي مثل تردد الراديو و GSM لتوفير المراقبة والتحكم عن بعد. يهدف النظام إلى أن يكون قابلاً للتطوير، وفعال من حيث التكلفة، وسهل التنفيذ، مما يجعله مناسبًا للاستخدام في مجموعة متنوعة من السياقات مثل المنازل والمكاتب والأماكن العامة. يوفر نظام الأمان المقترح حلاً موثوقًا وفعالًا لضمان السلامة والأمن في مجموعة متنوعة من المواقع من خلال الجمع بين تقنيات الذكاء الاصطناعي والإرسال اللاسلكي.

الكلمات المفتاحية:NVIDIA Jetson Nano ، الذكاء الاصطناعي، التعلم العميق، Arduino، GSM،وتردد الراديو.

List of tables

Table III. 1. Details about the difference between Yolov5 versions……..75

Table III. 2. Results of different evaluation criteria using YOLO models on the training and validation dataset… 83

Table IV. 1. NVIDIA Jetson Nano Card Technical Specifications. 94

List of figures

Figure I. 1. Examples of different scenes captured by CCTV: (a) a factory, (b) a crime, (c) a school, (d) a store, (e) a road, and (f) a home. 8
Figure I. 2. Analog CCTV Cameras Diagram. 10
Figure I. 3. IP CCTV cameras diagram. 10
Figure I. 4. Wireless CCTV camera. 11
Figure I. 5. CCTV system. 12
Figure I. 6. Access control systems. 13
Figure I. 7. Video surveillance system. 14
Figure I. 8. Intrusion detection systems. 15
Figure I. 9. Fire detection and suppression systems. 15
Figure I. 10. Perimeter protection systems. 15
Figure I. 11. Alarm systems. 16
Figure I. 12. Cyber security systems. 16
Figure I. 13. Radio frequency communication. 17
Figure I. 14. GSM Technologies. 17
Figure I. 15. Wireless Communication Technologies. 18
Figure I. 16. Diagram of a plane electromagnetic wave propagating in the X-axis. The electrical (E) and magnetic (H) components of the wave are represented in the Y and Z. 19
Figure I. 17. (a) Simplex RF communication system, (b) Half-duplex RF communication system, and (c) Full-duplex RF communication system. 20
Figure I. 18. Non line of sight communication. 21
Figure I. 19. GSM Architecture. 22
Figure I. 20. GSM Cell. 23
Figure I. 21. Hierarchical cell structure. 23
Figure I. 22. Cellular structure of the GSM network. 25

Figure II. 1. Relationship between Artificial intelligence, Machine Learning and Deep Learning. 30
Figure II. 2. Types of algorithms in machine learning. 31

Figure II. 3. Supervised learning workflow. 32
Figure II. 4. Unsupervised Learning. 33
Figure II. 5. (a) Biological neuron, and (b) Artificial neuron. 33
Figure II. 6. Fully Connected Feed Forward Multilayer Perceptron. 34
Figure II.7. Three activation functions: (a) Threshold, (b) Linear, and (c) Sigmoid. 35
Figure II. 8. Architecture of an RBF neural network. 36
Figure II. 9. (a) Separating Hyperplane, (b) Optimal Separating Hyperplane. 38
Figure II. 10. A schematic structure of SVM model. 38
Figure II. 11. Feature space is related to input space via a nonlinear map , causing the decision surface to be nonlinear in the input space. 39
Figure II. 12. Convolutional Neural Network. 41
Figure II. 13. The convolution operation. 42
Figure II. 14. The process of pooling operation. 43
Figure II. 15. Pictorial representation of ReLU functionality. 43
Figure II. 16. RELU function. 44
Figure II. 17. Sigmoid function. 45
Figure II. 18. Leaky Relu function. 45
Figure II. 19. Softmax function. 46
Figure II. 20. Tanh function. 47
Figure II. 21. AlexNet CNN architecture. 48
Figure II. 22. Basic layout of AlexNet architecture. 48
Figure II. 23. GoogLeNet inception model. 49
Figure II. 24. Illustration of convolutional layer used in MobileNet-v2 including depth wise separable convolutions, a shortcut connection, and linear bottleneck (hatched layers have no non-linearities). 49
Figure II. 25. Shortcut connection. 50
Figure II. 26. VGGNet Architecture. 51
Figure II. 27. VGG 16 Model. 51
Figure II. 28. VGG 19 Model. 51

Figure III. 1. Google Colab Notebook Setting. 62
Figure III. 2. Google Colab File Name. 62
Figure III. 3. Object Detection. 64

Figure III. 4. Classification with localization. 64
Figure III. 5. Categories of object detectors models. 65
Figure III. 6. A timeline of YOLO versions. 66
Figure III. 7. YOLO three step object detection process. 66
Figure III. 8. YOLO Architecture to Represent the Actions Performed on the Image. 67
Figure III. 9. Versions of YOLOv5. 68
Figure III. 10. The YOLOv5 architecture. 69
Figure III. 11. Illustration of intersection over union (IOU). 70
Figure III. 12. Illustration of IOU performance. 70
Figure III. 13. GIoU evaluation diagram. 71
Figure III. 14. Dataset samples for handguns Class and knifes class. 74
Figure III. 15. Connect to Google Colab Notebook to Google Drive. 76
Figure III. 16. Folder structure for dataset. 76
Figure III. 17. Cloning and installing the YOLOv5 repository. 77
Figure III. 18. Implement the training process. 77
Figure III. 19. Training progress in 100 epochs. 78
Figure III. 20. Use TensorBoard to load the entire training process saved in runs folder. 78
Figure III. 21. Implement the detection process. 79
Figure III. 22. Results of evaluation Metrics: (a) box loss, (b) the objectness loss, and (c) the classification loss. 80
Figure III. 23. Results of evaluation Metrics: (a) Recall, (b) Precision and Mean Average Precision. (c) mAP 0.5, and (d) mAP 0.5:0.95. 82
Figure III. 24. Detection results of weapons classes; handguns and knives for YOLOv5n. 83
Figure III. 25. Normal flow of intelligence video surveillance systems. 85

Figure IV. 1. Arduino Uno boards. 92
Figure IV. 2. Arduino Nano boards. 93
Figure IV. 3. NVIDIA Jetson NanoB01. 94
Figure IV. 4. Terminal Application. 95
Figure IV. 5. Connecting a UTP LAN cable to a NVIDIA Jetson Nano. 95
Figure IV. 6. 16×2 LCD display 97
Figure IV. 7. Buzzer. 97
Figure IV. 8. Access Point. 98

Figure IV. 9. (a) nRF24L01-PA-LNAModule, (b) nRF24L01 Module. 99
Figure IV. 10. nRF24L01 Radio frequency Transmitter/Receiver connection with the Arduino Uno board. 100
Figure IV. 11. Electrical circuit of nRF24L01 radio frequency Transmitter/Receiver Module with Arduino Uno Board. 101
Figure IV. 12. GSM/GPRS Shield. 103
Figure IV. 13. Connecting GMS Module with Arduino Uno. 104
Figure IV. 14. Electrical Circuit of the GSM Module with the Arduino Uno. 104
Figure IV. 15. The operation link between TE, TA, and ME. 106
Figure IV. 16. Camera Imou Bullet 2E. 107
Figure IV. 17. System Design Architecture and Connection. 108
Figure IV. 18. Transmitter Prototype. 110
Figure IV. 19. Power Supply. 110
Figure IV. 20. Connecting Components of Transmitter Section. 112
Figure IV. 21. Connecting the Nvidia Jetson Nano and the IP Camera. 112
Figure IV. 22. Configtool. 113
Figure IV. 23. Advanced IP Scanner. 113
Figure IV. 24. Remote Desktop Connection to the Nvidia Jetson Nano. 113
Figure IV. 25. ConnectingGSM900, Arduino Nano, and RF Transmitter with Arduino Uno. 114
Figure IV. 26. Nvidia Jetson Nano Login. 114
Figure IV. 27. Nvidia Jetson Nano, and Intelligent Monitoring System Files. 115
Figure IV. 28. Adding Files to Nvidia Jetson Nano by a USB flash. 115
Figure IV. 29. Adding sender code to Nvidia Jetson Nano. 115
Figure IV. 30. (a) Define Arduino Uno and the Logical Gate. 116
Figure IV. 31. Receiver Prototype. 117
Figure IV. 32. Choosing Yolov5 File. 119
Figure IV. 33. Opening yolov5 in Terminal. 119
Figure IV. 34. Weights File. 119
Figure IV. 35. Yolov5 Models. 120
Figure IV. 36. Nvidia Desktop. 120
Figure IV. 37. Detect Weapon Arduino File. 120
Figure IV. 38. Serial Communication between Nvidia Jetson Nan and Arduino Uno. 120

Figure IV. 39. (a) Detect Weapon Arduino file window, and (b) determine the type of detection. 121
Figure IV. 40. (a) Camera Interface, (b) Detection Interface. 122
Figure IV. 41. Project Prototype. 123
Figure IV. 42. (a) Example of Person Detection, (b) and (c) Person Interface Detection. .. 124
Figure IV. 43. (a) Example of weapon detection, (b) and (c) Weapon Interface Detection. 125
Figure IV. 44. GSM Alert. 125
Figure IV. 45. (a) Safety System, (b) Alert System. 125

Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
AuC	Authentication Center
ANN	Artificial Neural Networks
BS	Base Station
BTS	Base Transceiver System
BSC	Base Station Controller
BP	Backpropagation
CCTV	Closed Circuit Television
CPU	Central Processing Unit
COCO	Common Objects in Context
CNN	Convolutional Neural Network
CSP	Content Security Policy
DL	Deep Learning
DNN	Deep Neural Network
EIR	Equipment Identity Register
FN	False Negatives
FC	Fully Connected
GSM	Global System for Mobile Communication
GPU	Graphic Processing Units
GIoU	Gradient Intersection over Union
HLR	Home Location Register
IDS	Intrusion Detection Systems
IMSC	International Mobile Subscriber Identity
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
ISS	Intelligent surveillance system
IoU	Intersection over Union
ML	Machine Learning
MSC	Mobile Station Controller

MS	Mobile Station
MLP	Multilayer Perceptrons Network
MAP	Mean Average Precision
NS	Network Subsystem
NLP	Natural Language Processing
OSH	Optimal Separating Hyperplane
ONNX	Open Neural Network Exchange
RF	Radio Frequency
RBFNN	Radial Basis Function Neural Networks
RBF	Radial Basis Function
SIM	Subscriber Identity Module
SVM	Support Vector Machines
SSD	Single Shot Detector
TPU	Tensor Processing Unit
VGG	Visual Geometry Group
VLR	Visitor Location Register
YOLO	You Only Look Once

Table of Content

General Introduction 1
References cited in General Introduction 4

Chapter I

Smart Monitoring System

I.1. Introduction 7
I.2. General Information on CCTV 7
I.2.1. Definition of CCTV 7
I.2.2. Applications of CCTV 7
I.2.3. Characteristics of CCTV 9
I.2.4. Types of CCTV 9
I.2.5. How does CCTV Work? 11
I.3. Evolution of video surveillance systems 12
I.4. Current security technologies 13
I.5. Application of wireless communication technologies in security systems 17
I.5.1. Radio frequency communication 19
I.5.1.1. Advantages of radio frequency communication 21
I.5.2. GSM communications 21
I.5.2.1. Function of Components 22
I.5.2.2. Concept of Cellular Networks 23
I.5.2.3. GSM Network Structure 24
I.5.2.4. GSM Network Equipment 24
I.6. Conclusion 25
I.7. References cited in Chapter I 26

Chapter II

Artificial Intelligence

II.1. Introduction 30
II.2. Artificial Intelligence and Machine Learning 31
II.2.1. Supervised Learning 32

II.2.2. Unsupervised Learning 32
II.3. Artificial Neural Networks 33
II.3.1. Multilayer Perceptrons Network MLP 34
II.3.2. Radial Basis Function Neural Networks 36
II.4. Support Vector Machine SVM 37
II.4.1. Classification Mode 37
II.5. Deep Learning 39
II.5.1. Convolutional Neural Network 40
II.5.2. Several Common CNN Architectures 47
II.5.2.1. AlexNet 47
II.5.2.2. GoogLeNet 48
II.5.2.3. MobileNet 49
II.5.2.4. ResNet 50
II.5.2.5. VGG 50
II.6. Importance and Controversy of DL & AI 51
II.7. Conclusion 53
II.8. References cited in Chapter II 53

Chapter III

Simulation of Deep Learning Models

III.1. Introduction 59
III.2. Software aspects 59
III.2.1. Python 59
III.2.2. Tools and libraries 60
III.2.3. AI Accelerators 62
III.2.3.1. Centralized Processing Unit (CPU) 63
III.2.3.2. Graphical Processing Unit (GPU) 63
III.2.3.3. Tensor Processing Unit (TPU) 63
III.3. Object detection 63
III.3.1. YOLO 65
III.3.2. YOLOv5 model 68
III.3.2.1. YOLOv5 Architecture 69
III.3.2.2. Bounding-Box Regression and Loss Function 70

III.3.3. Evaluation Metrics 71
III.4. Methodology 72
III.4.1. Data collection 72
III.4.2. Dataset 73
III.4.3. Implementing YOLOV5 Algorithm 74
III.4.3.1. Yolov5 Setup 75
III.5. Results and discussions 79
III.6. Conception of an intelligent surveillance system 84
III.7. Conclusion 85
III.8. References cited in Chapter III 86

Chapter IV

Presentation and Realization of the Project

IV.1. Introduction 91
IV.2. Software and hardware components 91
IV.2.1. Processing Units 91
IV.2.1.1. Arduino boards 92
IV.2.1.2. Nvidia Jetson Nano Card 93
IV.2.2. Arduino shields 96
IV.2.2.1. LCD display 96
IV.2.2.2. Buzzer 97
IV.2.3. Wireless communication 97
IV.2.3.1. Access point 98
IV.2.3.2. nRF24L01 Modules 99
IV.2.3.2.1. Role of the nRF24L01 Modules in the Project 99
IV.2.3.3. GSM Module 102
IV.2.3.3.1. Connecting the GSM Module 103
IV.2.3.3.2. AT Commands 105
IV.2.3.3.3. Role of the GSM Module in the Project 106
IV.2.3.4. Camera IP Imou Bullet 2E 106
IV.3. Smart Security System 107
IV.3.1. Transmitter section 109
IV.3.1.1. Hardware Setup 109

IV.3.1.2. Connecting components 111
IV.3.2. Receiver Section 116
IV.3.2.1. Hardware Setup 117
IV.3.2.2. Transmission Protocol 118
IV.3.3. Video Processing 118
IV.4. Project Prototype 122
IV.5. Conclusion 126
IV.6. References cited in Chapter IV 127
General Conclusion 129

General Introduction

General Introduction

Technology has advanced at an exponential rate over the last few decades leading to significant advancements in artificial intelligence (AI) research. Today, AI has more prevalent than ever in a variety of global economic sectors. Additionally, as the economy and society continue to develop, the demand for efficient security systems increases.

In today's world, surveillance systems are commonly installed in public areas to ensure the safety of individuals and protect property. However, traditional security systems excessively rely on human resources, resulting in the cumbersome and time-consuming task of manually reviewing all recorded surveillance footage. To overcome these limitations and alleviate the burden of manual surveillance, the implementation of an intelligent security monitoring system is necessary. Such a system would not only address these challenges but also optimize the utilization of human resources, ultimately enhancing public security.

With the developments in deep learning and neural network, computer vision has now reached the point where it can recognize objects and patterns as well as the human eye. It is one of the tools aiding in the transition to a significant evolution in intelligent security systems.

To develop a smart security system, it is necessary to acquire knowledge about weapon detection and object detection. Our work has been inspired by various research studies in these areas. In the following, we will present and analyze several papers related to weapon detection and security, followed by an exploration of studies on object detection in a broader context.

In the domain of weapon detection, a research paper [1] conducts a comparative analysis of two state-of-the-art object detection algorithms, namely You Only Look Once (YOLO): YOLOv4 and YOLOv3, using a dedicated weapons dataset. Another paper [2] focuses on automatic gun and weapon detection, employing convolutional neural network (CNN) based SSD and Faster R-CNN algorithms. The study utilizes two types of datasets: pre-labeled images and manually labeled images. Additionally, to enhance video monitoring capabilities, a paper [3] presents the development of an intelligent video surveillance system. It first identifies the existing challenges in video surveillance and then proposes intelligent requirements in three specific areas.

Sharma and Mir [4] conducted a comprehensive survey that explores the recent advancements in deep learning-based object detection. The survey covers a wide range of topics, including an analysis of both traditional and current object detectors. It provides an overview of the

backbone architectures and learning strategies utilized in deep learning-based object detection. Additionally, the paper discusses popular datasets and metrics commonly employed in object detection research, offering valuable insights into the current landscape of object detection techniques.

In [5], the authors delve into object detection algorithms, with a specific focus on the CNN family and the YOLO approach. The paper begins by providing background information on CNNs, highlighting Faster R-CNN as an exemplary model. It then introduces two versions of YOLO, namely YOLO V1 and V2, and presents a comprehensive overview of the layers, algorithms, and characteristics of YOLO. A comparison between YOLO and Faster R-CNN is conducted, considering factors such as accuracy, speed, cost, and complexity, emphasizing the advantages and limitations of each approach.

The paper also acknowledges that the availability of appropriate training data poses a significant challenge in machine learning, as it plays a crucial role in achieving optimal results. Lastly, the paper concludes by summarizing the findings of YOLO and Faster R-CNN and provides a prognosis on the future of CNNs in the context of object detection.

Masita et al. [6] conducted a study that focuses on evaluating the performance of the R-CNN (Region-based Convolutional Neural Network) deep learning technique for pedestrian detection. The experiment involves the utilization of two distinct pedestrian detection datasets. The researchers employed the AlexNet deep learning feature extraction model in conjunction with an R-CNN detector. Moreover, the paper highlights the importance of considering the computational requirements of deep learning models. To determine the most suitable computing platform, additional experiments were conducted on various computing platforms. This investigation provides valuable insights into the performance and computational considerations of the R-CNN technique for pedestrian detection.

In [7], the document provides a comprehensive review of deep learning-based object detection algorithms specifically designed for challenging environments. It discusses recent approaches, performance analysis, advantages, and datasets used in these algorithms. The paper also addresses the current limitations of these approaches and highlights potential future research directions in the field. Zou et al. [8] present an extensive survey on object detection methods developed over the past two decades. The article describes major object detectors, providing a comprehensive overview of their characteristics and techniques. Furthermore, it offers valuable insights and suggests promising avenues for future research in the field of object detection.

The paper [9] offers an overview of the YOLOv5 model, emphasizing its architecture and key differences from previous versions, such as YOLOv3 and YOLOv4. It specifically focuses on the distinctions between YOLOv5 and the latest versions, namely YOLOv6 and YOLOv7. The paper compares various iterations of the YOLOv5 model using a common image dataset and provides detailed suggestions for researchers in choosing the most suitable model for a given problem.

Based on the papers discussed, we have selected the YOLOv5 model as the foundation for our work. This model has proven to be stable and has showcased remarkable performance in object detection tasks. Our research objective is to design and implement a smart security system that utilizes YOLOv5, to analyze real-time video streams and effectively detect weapons and individuals with high accuracy.

To ensure a prompt response and intervention, our system will incorporate wireless communication capabilities. This will enable the transmission of real-time alerts and notifications in emergency situations. In order to achieve real-time performance, we will employ powerful hardware resources such as GPUs, which are capable of handling the intensive computational requirements of the AI algorithm.

By combining advanced deep learning techniques, wireless communication, and robust hardware, we aim to develop a smart security system that significantly enhances public safety and security.

One of the major problems of our smart security system is to make the intelligent machine capable of understanding the environment in which it operates, specifically by detecting human presence and identifying dangerous situations such as weapon detection. Our work is divided into two phases. The first phase involves a calibration process aimed at determining the best parameters for our offline model. In the second phase, which operates in real-time, the system is capable of detecting and identifying dangers by leveraging the parameters extracted during the first phase. These two phases necessitate the essential application of artificial intelligence techniques for the automatic recognition of captured objects.

Another significant issue is the dependence on human intervention during emergency situations. With the emergence of wireless communication systems, we have harnessed these technologies to generate and transmit emergency messages. Notable examples of such systems include radiofrequency wireless communication and GSM technology. However, it is crucial to address the challenge of minimizing human involvement in the identification of dangerous situations and establish efficient and automated mechanisms for emergency cases.

In order to achieve the underlined goal of our work, we have divided this dissertation into the following four chapters as follows:

The first Chapter **‘‘***Smart Monitoring System* ’’ explores smart monitoring systems, including CCTV and video surveillance. It discusses the growth of video surveillance and the use of wireless communication technologies in security systems.

The second Chapter **‘‘***Artificial Intelligence* ’’ introduces artificial intelligence, and machine learning. In particular, it focuses on the fundamentals of deep learning, especially convolutional neural networks (CNNs). The entire notion forms the basis for the implementations described in the next chapter.

The third chapter ‘‘*Simulation of Deep Learning Models*’’ focuses on simulating deep learning models and selecting the best models for future real-time applications, specifically for weapons detection.

The fourth Chapter ‘‘*Practical Realization’’* describes the realization of the system, using NVIDIA card and wireless communications technologies. It highlights the implementation process and showcases the integration of hardware and software components to create a smart security system.

References cited in General Introduction

[1] T. S. S. Hashmi, N. U. Haq, M. M. Fraz, and M. Shahzad, "Application of deep learning for weapons detection in surveillance videos," in *2021 international conference on digital futures and transformative technologies (ICoDT2)*, 2021, pp. 1-6.

[2] H. Jain, A. Vikram, A. Kashyap, and A. Jain, "Weapon detection using artificial intelligence and deep learning for security applications," in *2020 International conference on electronics and sustainable communication systems (ICESC)*, 2020, pp. 193-198.

[3] C. She, R. Zheng, Q. Yang, and S. Liang, "Research of Intelligent Video Surveillance System based on Artificial Neural Network," in *Journal of Physics: Conference Series*, 2022, p. 012057.

[4] V. Sharma and R. N. Mir, "A comprehensive and systematic look up into deep learning based object detection techniques: A review," *Computer Science Review,* vol. 38, p. 100301, 2020.

[5] J. Du, "Understanding of object detection based on CNN family and YOLO," in *Journal of Physics: Conference Series*, 2018, p. 012029.

[6] K. L. Masita, A. N. Hasan, and S. Paul, "Pedestrian detection using R-CNN object detector," in *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2018, pp. 1-6.

[7] M. Ahmed, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Survey and performance analysis of deep learning based object detection in challenging environments," *Sensors,* vol. 21, p. 5116, 2021.

[8] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE,* 2023.

[9] M. Horvat and G. Gledec, "A comparative study of YOLOv5 models performance for image localization and classification," in *Central European Conference on Information and Intelligent Systems*, 2022, pp. 349-356.

Chapter I

Smart Monitoring System

I.1. Introduction

Smart monitoring systems are revolutionizing the way we collect and analyze data. By utilizing sensors, cameras, and other devices, these advanced technology systems can monitor a wide range of environments, infrastructure, and activities. For example, in the industrial automation sector, smart monitoring systems are being used to monitor production lines, detect equipment failures, and optimize workflow. In the security and surveillance industry, these systems are helping to prevent crime by detecting and alerting users to suspicious activity in real-time. The benefits of using smart monitoring systems are numerous, including improved safety, efficiency, and cost savings. As these systems continue to evolve and become more sophisticated, their potential applications will only continue to expand.

Surveillance systems, specifically CCTV (Closed Circuit Television) cameras, are ubiquitous in modern society, with installations in homes, shopping malls, schools, offices, and public spaces. In this chapter, we will explore the different types of CCTV cameras, their applications, and how they work. We will also examine the evolution of video surveillance systems, and how these technologies have developed over time.

We'll then dive into the latest developments in the field of security systems. As wireless communication technologies are an integral part of modern security systems, we'll take a closer look at radio frequency and GSM technologies and how they are being applied to enhance the effectiveness of these systems.

I.2. General Information on CCTV

I.2.1. Definition of CCTV

CCTV is an abbreviation for "Closed Circuit Television" and refers to a network of surveillance cameras used for monitoring and security.

These cameras are typically linked to a closed network of monitors and recording devices that security professionals use to monitor and record activity in a specified area. CCTV systems are commonly used in public places such as banks, shopping malls, airports, and government buildings, as well as in private businesses and homes. They are designed to deter illegal behavior, provide evidence for criminal investigations, and improve the safety and security of persons and property [1-3].

I.2.2. Applications of CCTV

CCTV systems have a wide range of applications and are used in various settings for different purposes. As shown in Figure I.1, here are some common uses of CCTV [4, 5]:

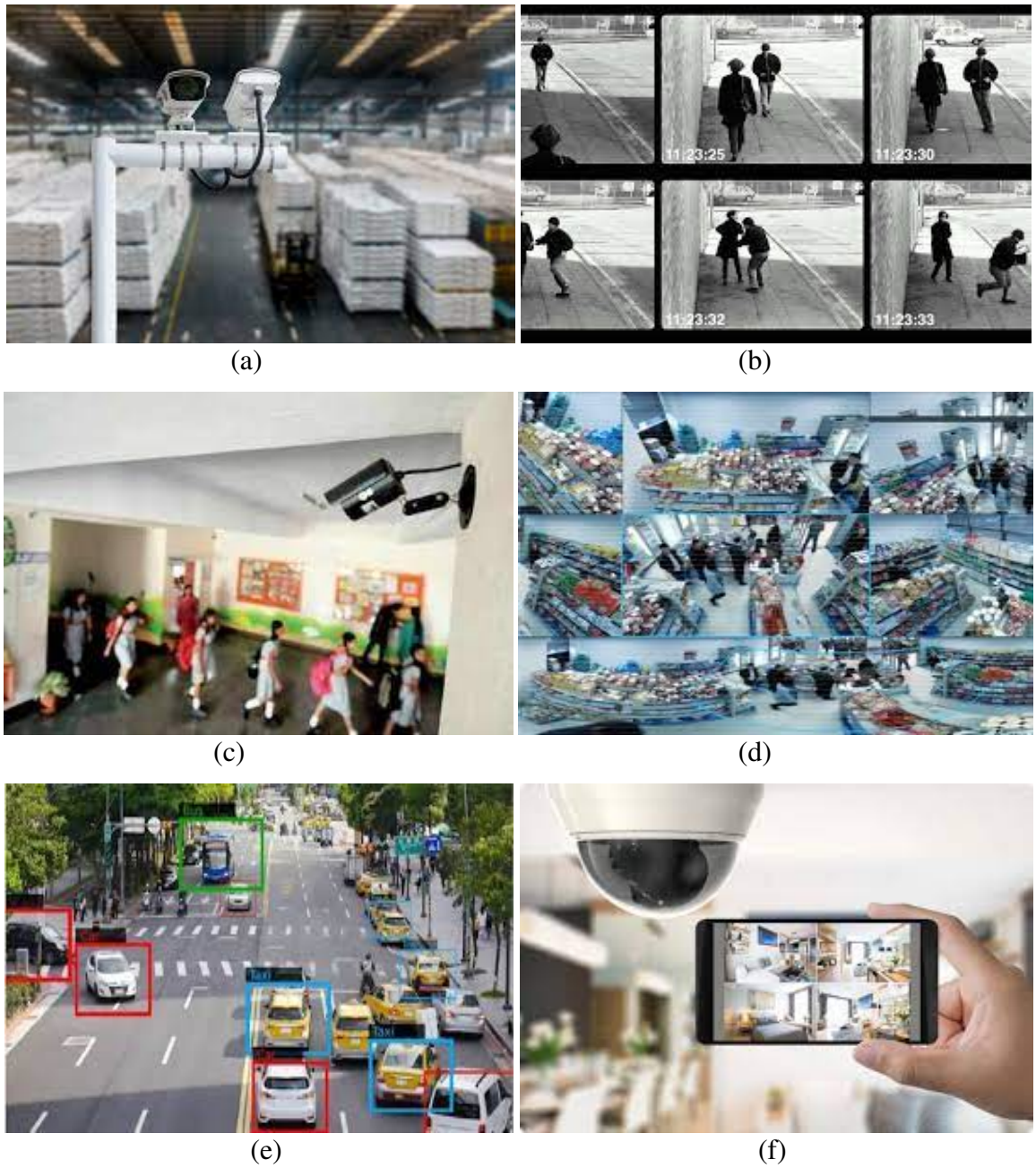


Figure I. 1. Examples of different scenes captured by CCTV: (a) a factory, (b) a crime, (c) a school, (d) a store, (e) a road, and (f) a home.

- Crime prevention and detection: CCTV cameras are frequently employed to dissuade criminal conduct as well as to give evidence in the event of a crime. Also, it can be used to monitor both public and private environments such as streets, parks, and shopping districts.
- Industrial and manufacturing: CCTV systems are used to monitor production processes, detect equipment malfunctions or breakdowns, and increase safety and efficiency in factories and manufacturing plants.

- Retail: we use CCTV cameras to monitor client behavior and prevent theft and larceny.
- Transportation: CCTV systems are used to monitor passenger activities, detect security concerns, and protect public safety in airports, train stations, and bus terminals.
- Education: we use it in schools and universities to monitor student behavior and ensure campus safety.
- Home security: we use theme in private homes to monitor and protect property, deter intruders, and provide evidence in the event of a break-in.

I.2.3. Characteristics of CCTV

Here are some common characteristics of CCTV systems [6, 7]:

- ✓ Video cameras: It's used for recording images and videos of the area under surveillance. These cameras can be fixed or remotely manipulated to vary their viewing angle.
- ✓ Recording devices: The video cameras views and recordings are recorded and saved on recording devices such as DVRs (digital video recorders) or NVRs (network video recorders).
- ✓ Monitors that display live or recorded images from the cameras are also included in CCTV systems. These monitors can be found in a central control room or scattered around the monitored area.
- ✓ Transfer video footage from cameras to recording and monitoring devices using a variety of transmission technologies. Wired systems, such as coaxial cables or fiber optic cables, and wireless systems, such as Wi-Fi or cellular networks, are examples of these.
- ✓ Some advanced systems of CCTV analyze video footage using analytics software to detect specific behaviors or events. For example, the software may recognize when someone enters a restricted area or removes an object off a shelf.
- ✓ Integration with other security systems: Those systems can be combined with other security systems such as access control, alarm, and fire detection systems to give a complete security solution.

I.2.4. Types of CCTV

There are several types of CCTV systems available, each with its own characteristics and uses. The following text describes some of the most common types [3, 8, 9]:

- o Analog CCTV: This is the traditional type of CCTV system that captures and stores video footage using analog cameras and recording equipment, as shown in Figure I.2 [10].

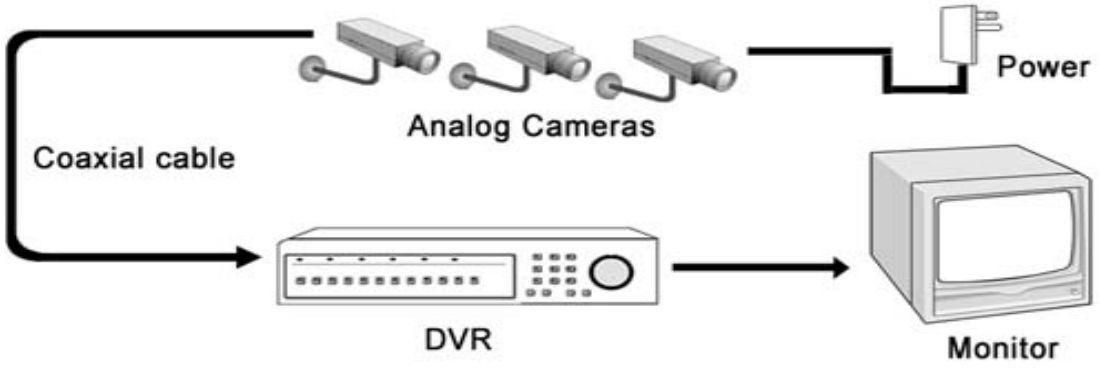


Figure I. 2. Analog CCTV Cameras Diagram.

- Digital CCTV: Capture and save video footage with digital cameras and recording devices. In comparison to analog systems, this type of system can provide higher resolution and more advanced features.
- IP CCTV: As illustrated in Figure I.3, stands for internet protocol CCTV systems, it includes digital cameras that are linked to an IP network, allowing video footage to be watched and stored remotely [11].

Figure I. 3. IP CCTV cameras diagram.

- Wireless CCTV: Transport video footage from cameras to recording and monitoring equipment via wireless technology. This type of system may be less difficult to establish and more adaptable than wired systems, as shows in Figure I.4 [12, 13].

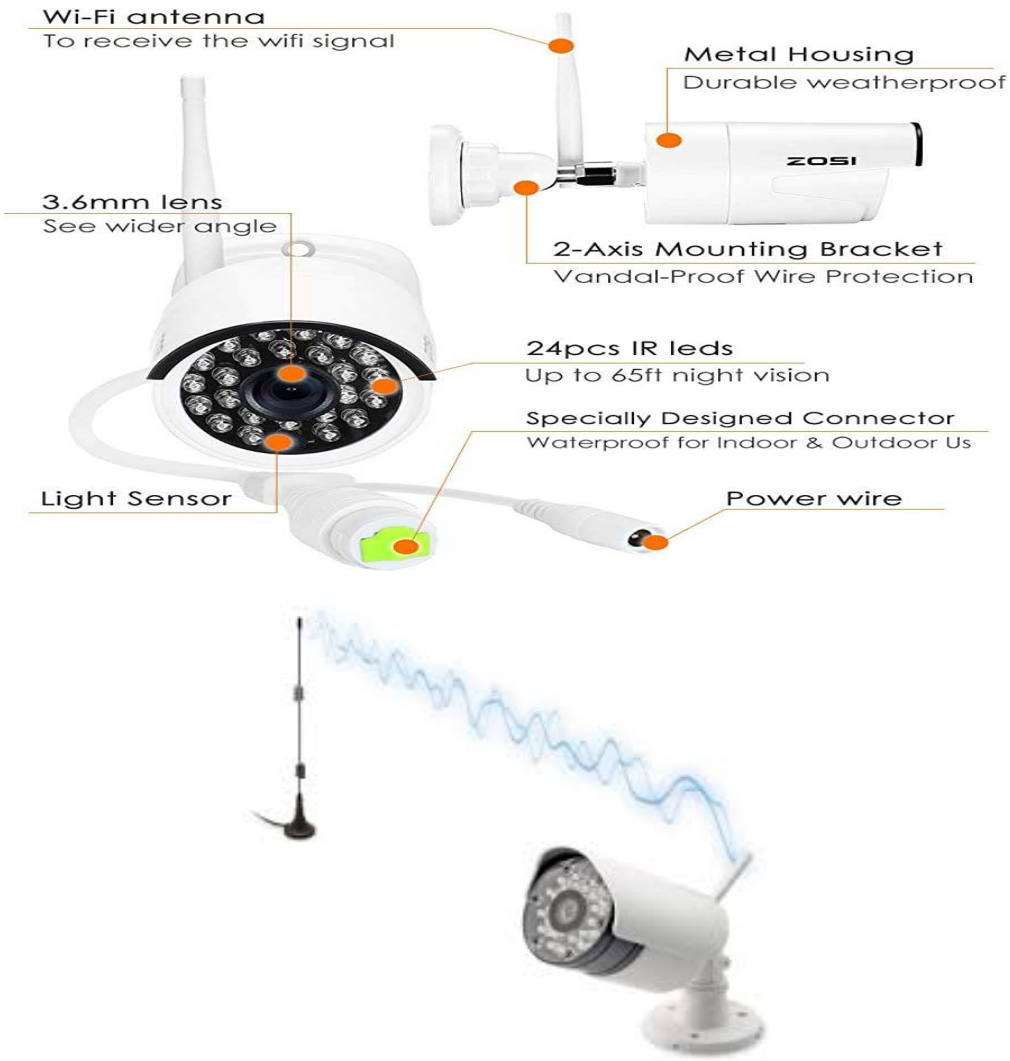


Figure I. 4. Wireless CCTV camera.

- Networked CCTV: To provide broad coverage of an area, it use several cameras connected to a network. These systems can also be combined with other security systems to provide a comprehensive security solution.
- HD CCTV: Stands for High Definition CCTV, these systems provide high-quality video footage by utilizing high-resolution cameras and recording technology. Also this type of system can be beneficial in applications requiring precise details, such as facial recognition or license plate recognition.

I.2.5. How does CCTV Work?

CCTV works by using cameras to capture video footage of a specific area, which is then transmitted to a limited set of monitors or recording devices in a closed circuit. The cameras can be connected to a video signal combiner, which is then connected to one or more video generating means and a video storage means in series or in parallel through a single cable to

simplify a connection there between [14]. The video footage can be stored for later viewing or monitored in real-time by human operators in a control room [15]. The effectiveness of CCTV systems can depend on various factors, including the design of the control room and the expectations and perceptions of the people being monitored [16]. As shown in Figure I.5, a CCTV system [17]:

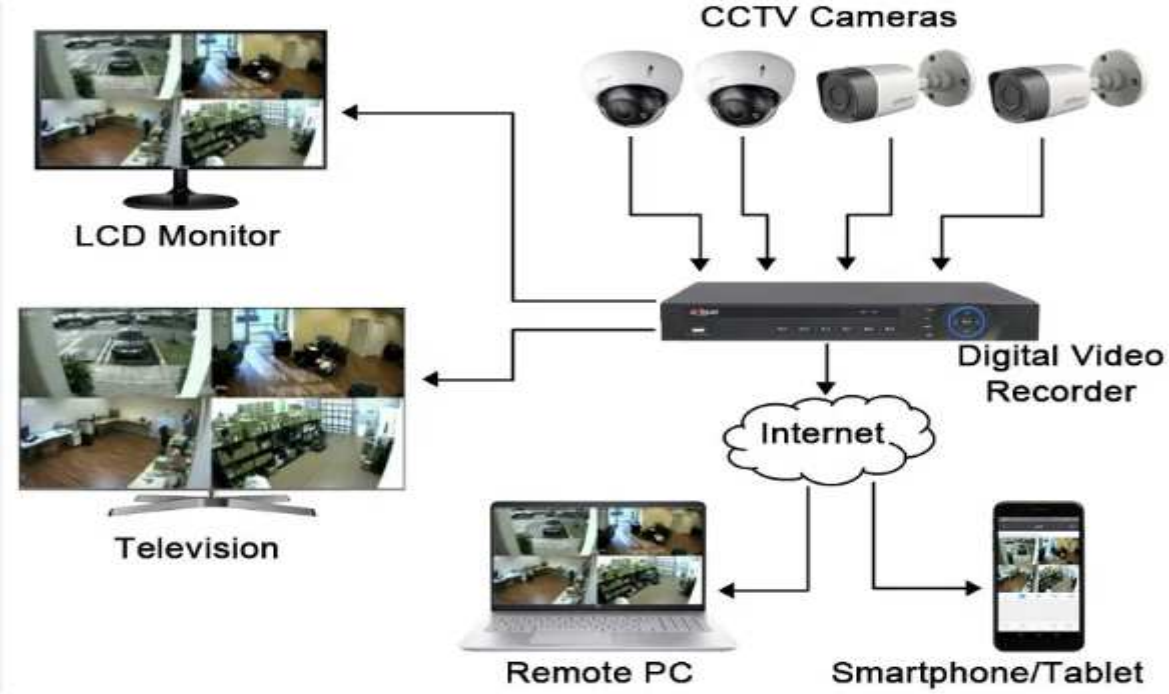


Figure I. 5. CCTV system.

I.3. Evolution of video surveillance systems

Video surveillance systems, also known as closed-circuit television (CCTV) systems, are used to monitor and record video footage in a particular area. These systems typically consist of cameras that capture video footage, recording equipment that stores the footage, and monitoring equipment that allows security personnel to view the footage in real-time. Video surveillance systems have evolved significantly over the years to meet the increasing demand for sophisticated surveillance systems.

Closed-circuit television (CCTV) was invented in the early 1940s to monitor the launch of V-2 rockets in Germany. CCTV systems, however, did not become widely used for security purposes until the 1960s and 1970s, particularly in commercial and industrial settings.

- The first generation of CCTV systems relied on cumbersome analog cameras and recording devices with limited storage capacity. To monitor and safeguard a property, they

were frequently employed in conjunction with other security measures, such as security guards.

- The advancement of digital video technology in the 1980s and 1990s resulted in the second generation of video surveillance systems. Digital cameras and recorders improved storage capacity and video quality, making it easier to collect and preserve vast volumes of material. These systems were also more adaptable and scalable, with the ability to add cameras and storage as needed.
- With the introduction of networked cameras and digital video recorders in the early 2000s, the third generation of video surveillance systems arose. These systems provided greater remote access and control, allowing for easier monitoring and management of several sites from a single location. They also allowed video surveillance to be integrated with other security systems, such as access control and alarm systems.

With the integration of analytics and artificial intelligence technologies, video surveillance systems are getting increasingly advanced. These systems can detect and inform security personnel to possible threats automatically, as well as give real-time insights about security and operational performance [18-25].

I.4. Current security technologies

Current security technologies include a wide range of tools and techniques that are used to protect people, assets, and infrastructure from various threats. Some of the most common security technologies include [26-28]:

- Access Control Systems: These systems regulate who has access to specific places or resources. They can include biometric devices like fingerprint scanners, as well as card readers and other forms of authentication (see Figure I.6) [29].

Figure I. 6. Access control systems.

➢ Video Surveillance: As previously noted, video surveillance systems are frequently utilized to watch and record behavior in a number of settings (Figure I.7) [30].

Figure I. 7. Video surveillance system.

➢ Intrusion Detection Systems (IDS): As illustrated in Figure I.8, these systems identify and notify security staff of unwanted access or attempted breaches [31].

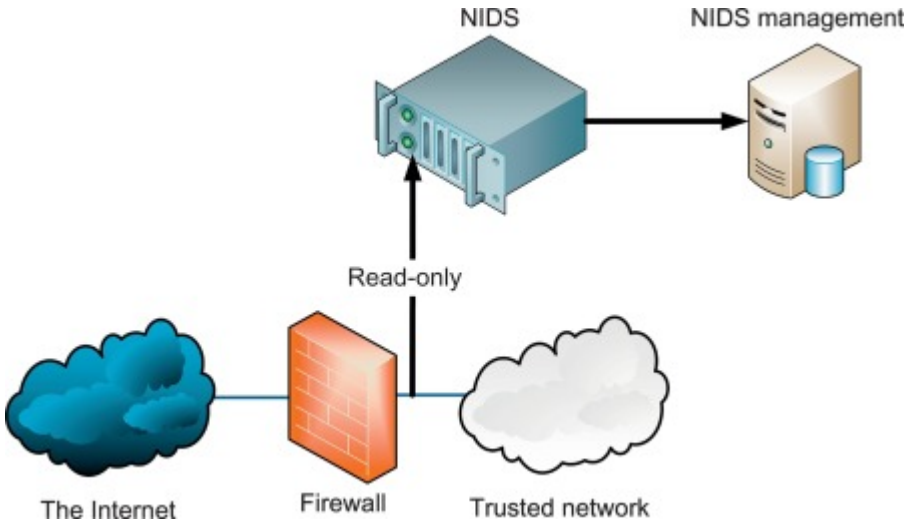


Figure I. 8. Intrusion detection systems.

➢ Fire detection and suppression systems: These systems detect and respond to fires, either by warning residents or by automatically activating fire suppression devices (see Figure I.9) [32].

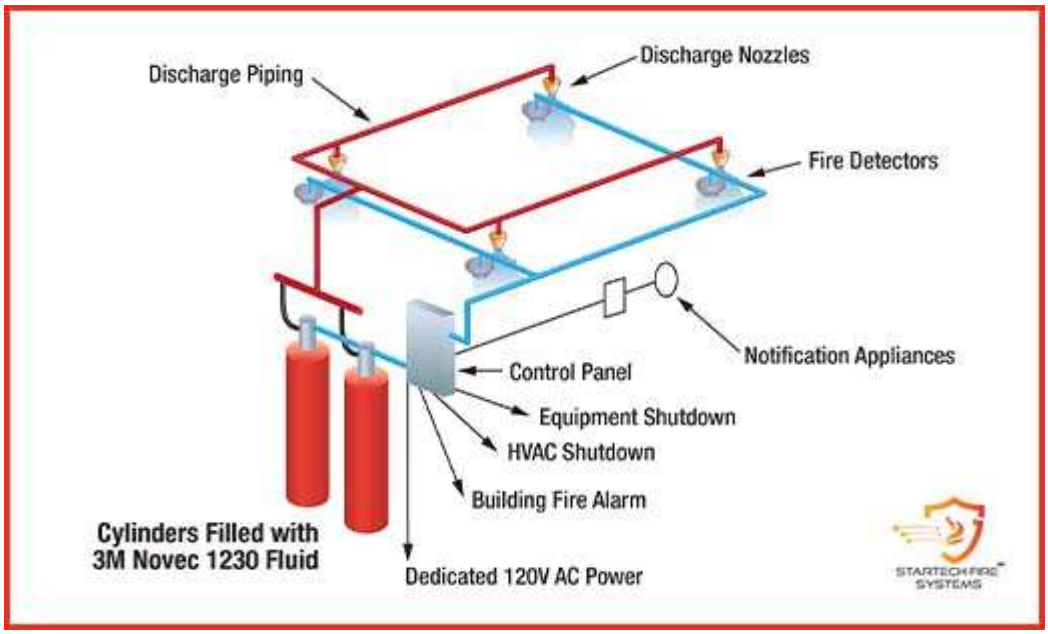


Figure I. 9. Fire detection and suppression systems.

➢ Perimeter protection: These technologies are used to protect a property's or facility's perimeter. Fences, barricades, and other physical security measures can be included (Figure I.10) [33].

Figure I. 10. Perimeter protection systems.

➢ Alarm systems: These systems warn security staff or building inhabitants to potential threats or crises (see Figure I.11) [34].

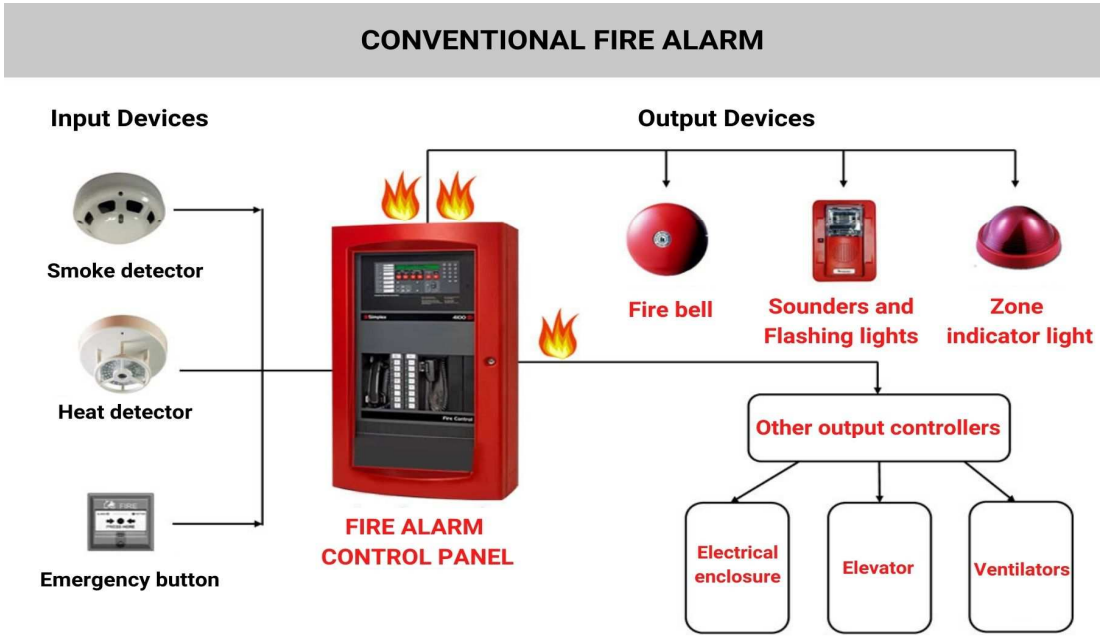


Figure I. 11. Alarm systems.

➢ Cyber security: These technologies are used to safeguard against digital risks like hacking, malware, and other cyber attacks. Artificial Intelligence (AI), Machine Learning (ML), Cloud Security, Block chain, and the Internet of Things (IoT) are some of the most recent cyber security technologies, as shown in Figure I.12 [35].

Figure I. 12. Cyber security systems.

Overall, security technologies are getting more advanced and integrated, enabling for more effective and efficient defense against a wide spectrum of threats.

I.5. Application of wireless communication technologies in security systems

Wireless communication refers to the transmission of data, information or signals without the use of physical cables or wires. Some of the most common wireless communication technologies (see Figure I.13) [36] used in security systems include:

- Radio frequency (RF) communication: uses radio waves to wirelessly transport data between devices. It is widely employed in wireless security systems, such as wireless alarm systems, and can send signals over quite large distances [37].

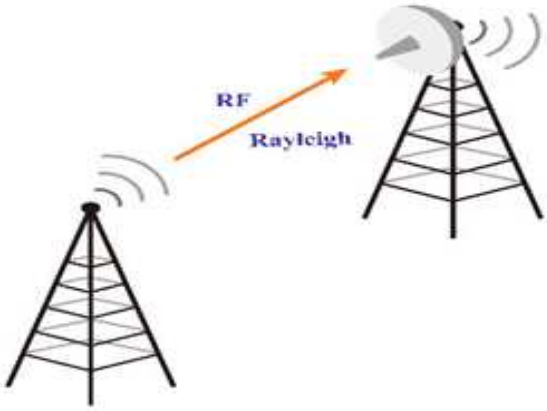


Figure I. 13. Radio frequency communication.

- Global system for mobile communication (GSM): is a wireless communication standard that is utilized by mobile devices such as smart phones. As shown in Figure I.14, it can also be utilized in security systems to enable wireless communication [38].

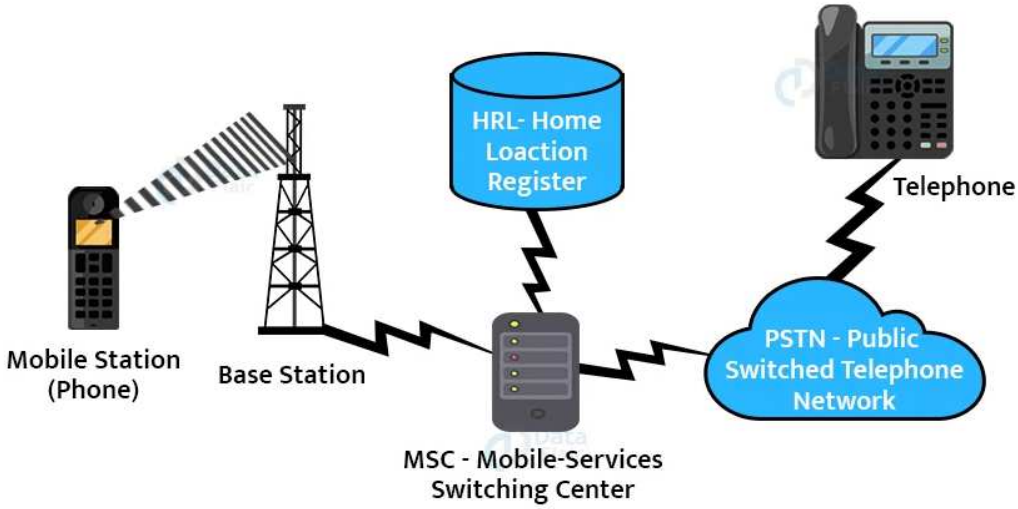


Figure I. 14. GSM Technologies.

- Wi-Fi: is a widely used wireless networking technology that enables devices can connect to the internet and communicate with one another without the use of physical connections. It is widely employed in video surveillance systems and other security applications.
- Bluetooth: is a wireless communication technology used in security systems to facilitate wireless communication between equipment such as sensors and alarms [39, 40].

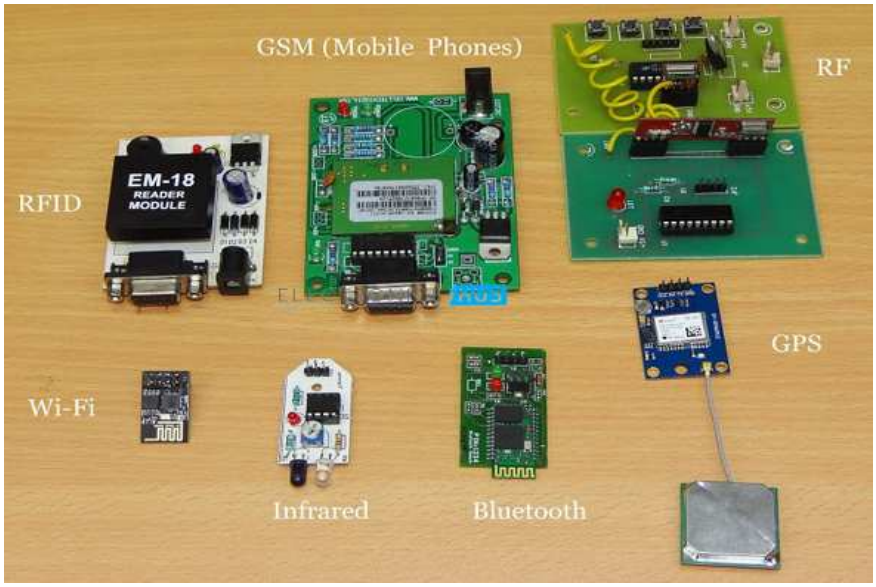


Figure I. 15. Wireless Communication Technologies.

Wireless communication can be used for a wide range of applications, including telecommunications, networking, broadcasting, and security systems, among others. Moreover, in security systems, wireless communication technologies have numerous applications that provide enhanced flexibility, scalability, and efficiency, making them an ideal choice for many security applications. Some common applications of wireless communication technologies in security systems include:

- ❖ Wireless alarm systems: Communicate signals between sensors and the main control panel using wireless communication technologies such as RF or GSM. This allows the system to be rapidly and easily setup without the need for substantial cabling.
- ❖ Video surveillance systems: Wi-Fi and Bluetooth wireless communication technologies can be utilized to relay video feeds from cameras to a central monitoring station. This enables remote monitoring and can improve coverage in regions where cabling is impractical.

- ❖ Access control systems: Wireless communication technologies, such as RFID, can be utilized for access control systems, allowing for secure and efficient access management without the usage of physical keys or cards.
- ❖ Environmental monitoring systems: It can be used to monitor environmental factors like temperature, humidity, and air quality, providing real-time data to assist keep the environment safe and healthy [41, 42].

In our Master's thesis, we focused on GSM and RF communication. In subsections I.5.1 and I.5.2, we introduce two popular types of wireless communication technologies: radio frequency communication (RF) and GSM communication. These technologies are widely used in various security systems applications.

I.5.1. Radio frequency communication

The electromagnetic wave or radiation consists of electric (E) and magnetic (H) fields whose alternation over time ensures propagation in space by a phenomenon of mutual induction (Figure I.16). Depending on the distance to the transmitting antenna and according to the relative dimensions of the latter in relation to the wavelength of the radiation [43].

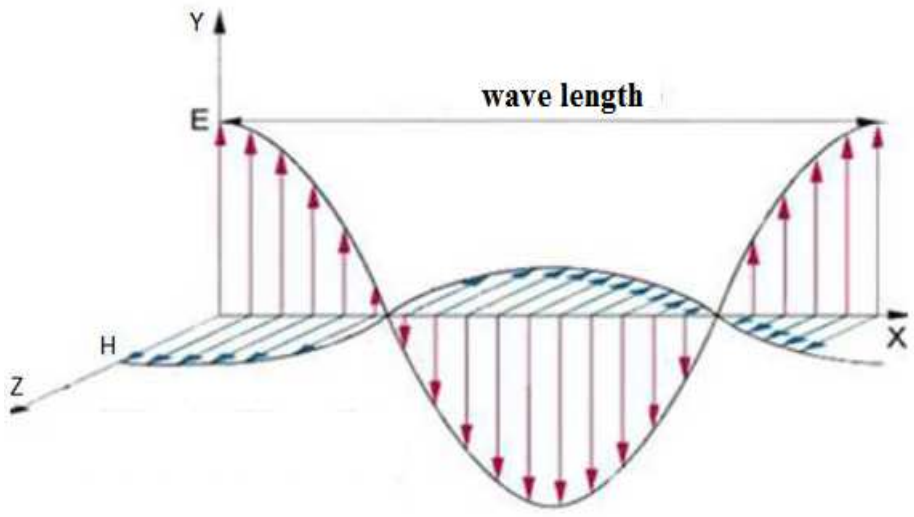


Figure I. 16. Diagram of a plane electromagnetic wave propagating in the X-axis. The electrical (E) and magnetic (H) components of the wave are represented in the Y and Z.

Radio frequencies (RF) are waves belonging to the electromagnetic spectrum and which range, in terms of frequency, from around 100 kHz to around 10 GHz. This band frequency is used by the radio frequency antennas (example: shield nRF24, with a 2.4GHz frequency), mobile phones, Wi-Fi, microwave ovens, etc. A wave RF is, like any electromagnetic wave,

composed of an electric and magnetic field that oscillate in phase, perpendicular to each other and perpendicular to the direction of propagation.

We have three systems of RF communication that are illustrated in Figure I.17, namely:

- ✓ Simplex RF system: Radio technology that allows only one-way communication from a transmitter to a receiver [44, 45]. As examples, we can mention: FM radio, Pagers, TV, One-way AMR systems.
- ✓ Half-duplex RF system: Operation mode of a radio communication system which each end can transmit and receive, but no simultaneously [44, 46]. As examples: Walkie-talkie, Wireless keyboard mouse.
- ✓ Full-duplex RF system: Radio system in which each end can transmit and receive simultaneously [44, 47]. For examples of the third system: Cellular phones, Satellites communication.

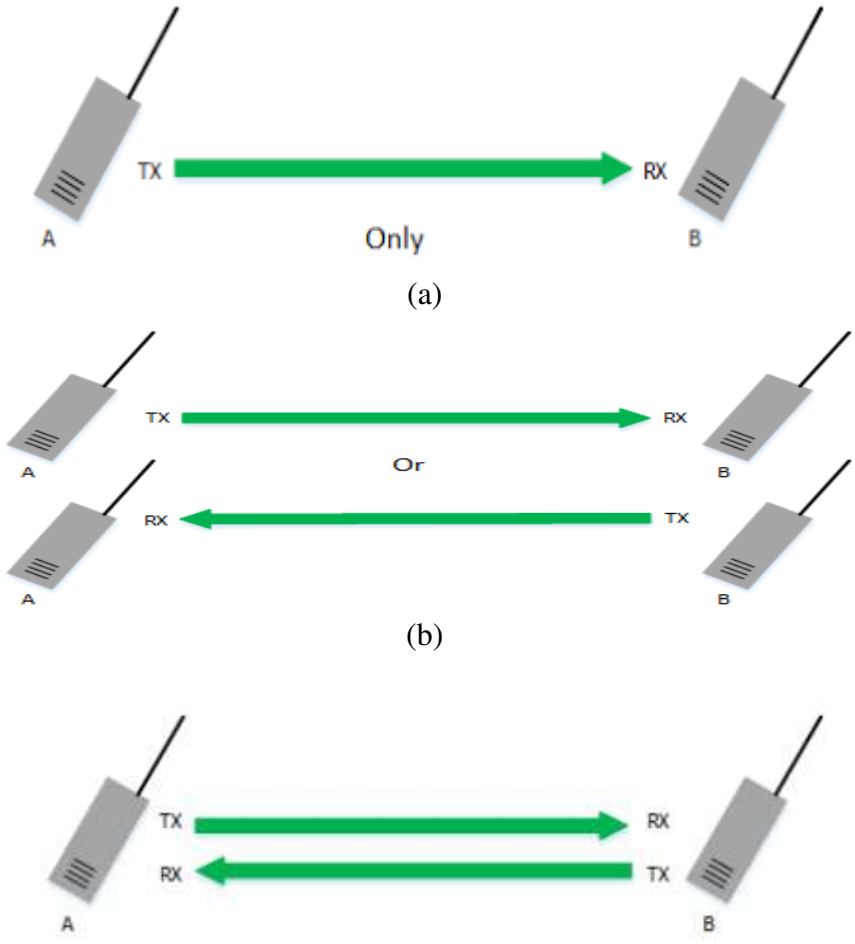


Figure I. 17. (a) Simplex RF communication system, (b) Half-duplex RF communication system, and (c) Full-duplex RF communication system.

I.5.1.1. Advantages of radio frequency communication

RF communication technology has several advantages, including:

- ✓ Non-Line-of-Sight (NLOS) communication: RF waves can penetrate walls, obstacles and other barriers, allowing communication to take place even when direct line-of-sight is not possible (Figure I.18).

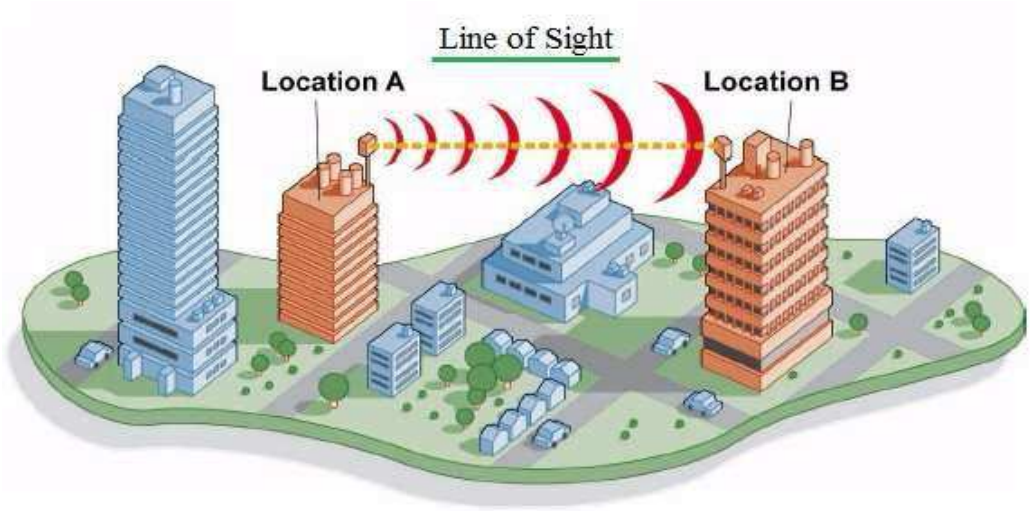


Figure I. 18. Non line of sight communication.

- ✓ Low power consumption: RF communication typically requires low power consumption, making it suitable for battery-powered devices.
- ✓ Low cost: RF communication is a relatively low-cost technology compared to other wireless communication technologies, making it ideal for low-cost security systems.
- ✓ Simple implementation: RF communication technology is easy to implement, and requires little infrastructure or equipment.

Some common applications of RF communication in security systems include wireless sensors, remote controls, and alarm systems. RF communication is also used in keyless entry systems, remote car starters, and other similar applications.

However, RF communication also has some limitations, including limited range, susceptibility to interference, and security vulnerabilities. Despite these limitations, RF communication remains an important technology in security systems and is likely to continue to play an important role in the future of wireless communication.

I.5.2. GSM communications

GSM (Global System for Mobile Communications) is a digital cellular technology used for mobile communication. It is one of the most widely used communication technologies in the world and is used by billions of people to make voice calls, send text messages, and access

the internet through their mobile devices. GSM was first introduced in 1991 and has since undergone numerous upgrades and improvements. It is based on a system of digital radio cells, which enables mobile devices to communicate with a network of base stations, and is capable of supporting high-speed data transfer and other advanced features. GSM technology is used by a large number of mobile operators around the world and is considered a standard for mobile communication in many regions.

The functional architecture of GSM can be divided between the Mobile Station (MS), the Base Station (BS) and the Network Subsystem (NS). The MS is managed by the user, the BS subsystem controls the radio link with the MS, and the NS switches calls between the mobile and the others which are fixed or the users of the mobile network as well as Mobility Management (see Figure I.19) [48].

I.5.2.1. Function of Components

Mobile station (MS): It refers for mobile station. Simply, it means a mobile phone.

- Base transceiver system (BTS): It maintains the radio component with MS.
- Base station controller (BSC): Its function is to allocate necessary time slots between the BTS and MSC.
- Home location register (HLR): It is the reference database for subscriber parameters like subscriber's ID, location, authentication key, etc.
- Visitor location register (VLR): It contains a copy of most of the data stored in HLR, which is temporary and exists only until the subscriber is active.
- Equipment identity register (EIR): A database contains a list of valid mobile equipment on the network.
- Authentication center (AuC): It performs authentication of subscribers.

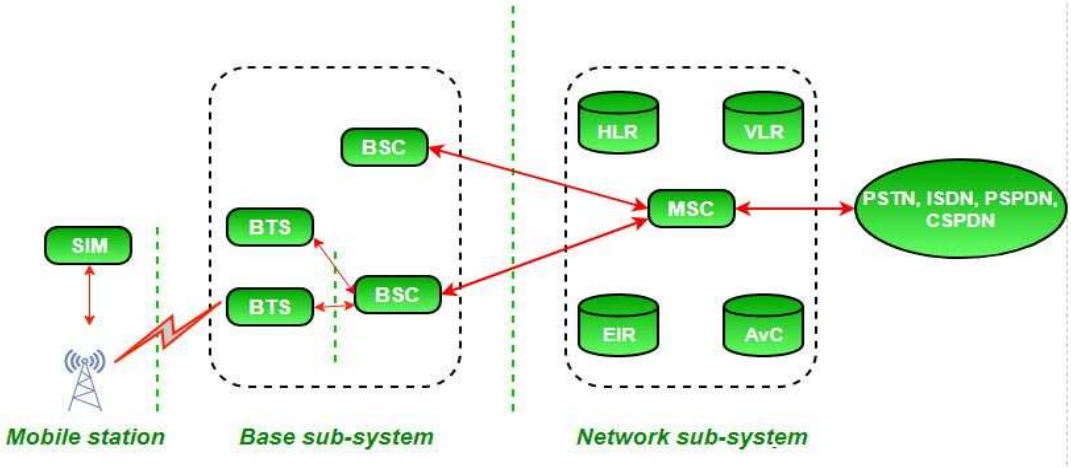


Figure I. 19. GSM Architecture.

I.5.2.2. Concept of Cellular Networks

In a GSM network, the territory is divided into small areas called cells, as shown in Figure I.20. Each cell is equipped with a fixed base station equipped with its antennas installed on a high point (water tower, building, etc.).

The cells are drawn hexagonal but the actual range of the stations depends on the configuration of the watered territory and the radiation pattern of the transmitting antennas. In practice, the cells therefore partially overlap [49].

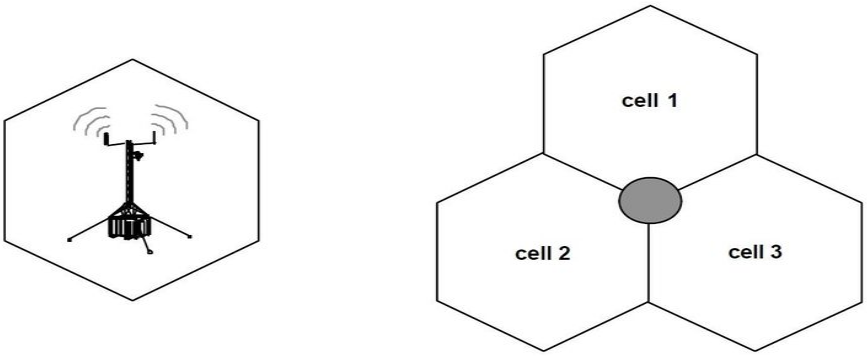


Figure I. 20. GSM Cell.

As shown in the following figure, the GSM cell size, there are four hierarchical levels of cells [50]:

- Pico-cell covering a small area such as the interior of an office.
- Micro-cell covering the surface of a small city.
- Macro-cell that can have coverage of several kilometers.
- Global cell covering a region that can reach one third of the gobe thanks to the satellites.

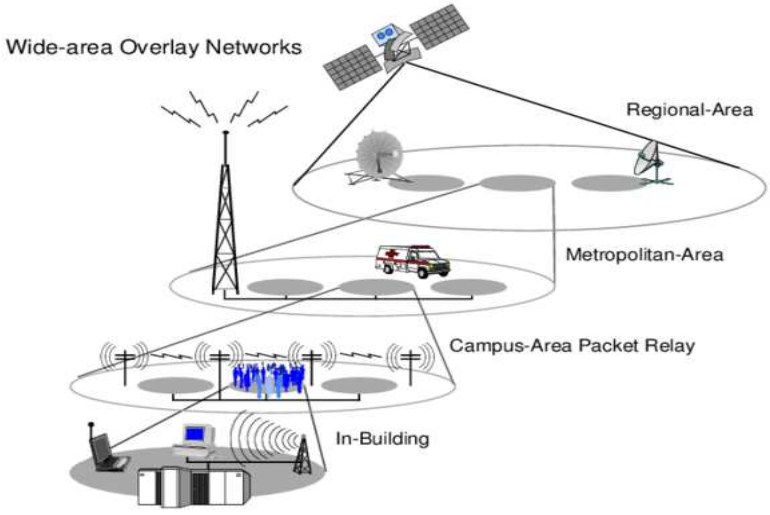


Figure I. 21. Hierarchical cell structure.

I.5.2.3. GSM Network Structure

When calling from a GSM mobile:

- ✓ The mobile transmits the communication by radio to the base station of its cell.
- ✓ The conversation is then routed in a more traditional way (cable, optical fiber, etc.)
- ✓ To the correspondent if he is connected to the wired telephone network, or to his base station.
- ✓ Base if equipped with a mobile.
- ✓ This base station finally transmits the conversation by radio to the correspondent.

Even if two people are in the same cell and are on the phone, the conversations never go directly from one mobile phone to another. During a movement, it is possible to leave a cell. It is then necessary to change the base station while maintaining communication: this is the transfer intercellular or handover.

To manage this transfer:

- ❖ The GSM phone continuously measures the strength of the radio signal received from the base station and regularly listens to the base stations of neighboring cells.
- ❖ When he notices that he receives better another base station than the one with which it exchanges the signals, it informs its base station.
- ❖ The base station then decides to pass the relay to the neighboring base station and puts in implements the handover procedure.

This process forces all GSM mobiles to listen to the base stations of neighboring cells in addition to the base station of the cell in which it is located.

I.5.2.4. GSM Network Equipment

The functions implemented in the GSM network are those required in any network of mobile devices such as dialing, routing to a mobile user, transfer of cells, etc…[51].

As shown in the Figure I.22, the GSM telephone or mobile station is characterized by two identities:

- ✓ The equipment number, IMEI (International Mobile Equipment Identity) put in the memory of the mobile during its manufacture.
- ✓ The IMSI (International Mobile Subscriber Identity) subscriber number found in the subscriber's SIM (Subscriber Identity Module) card.
- ✓ The radio communication system is the equipment that provides coverage of the cell and includes:
- ✓ BTS (Base Transmitter Station) base transmission stations.

- ✓ The base station controller BSC (Base Station Controller) which manages between 20and 30 BTS and has its register of visitor subscribers VLR storing the subscriber information related to his mobility.
- ✓ The mobile services switch MSC is an automatic switch that performs the functions of necessary switching by directing the conversations to the correspondent's MSC or to other networks (telephone, internet, etc.) through interfaces appropriate.
- ✓ The register of nominal subscribers or HLR (Home Local Register) is a database used for managing mobile subscribers and containing two types of information:
- ✓ Subscriber information, subscriber number (IMSI).
- ✓ Subscriber location information, allowing incoming calls to the network to be routed to this mobile.

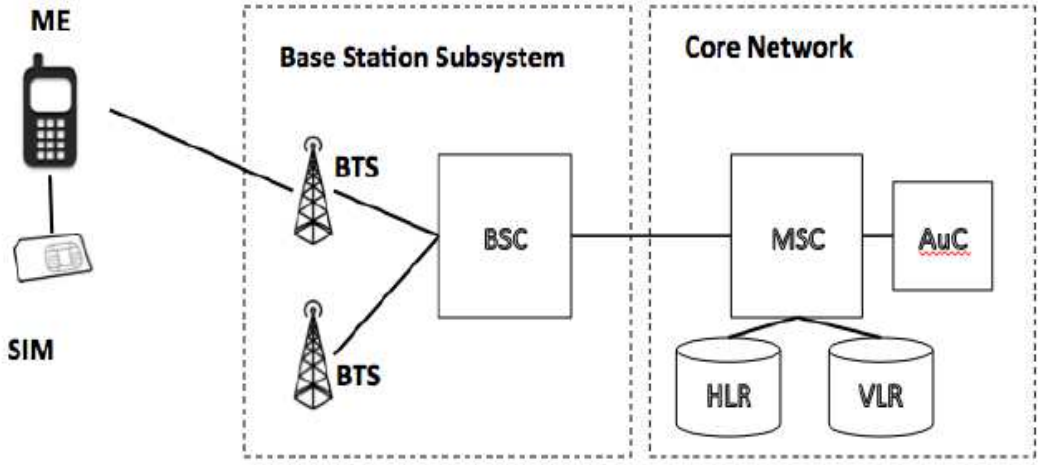


Figure I. 22. Cellular structure of the GSM network.

I.6. Conclusion

In this chapter, we provided an overview of existing security systems and discussed two different wireless communication technologies: radio frequency communication and the GSM system.

One of the key takeaways from this chapter is that traditional CCTV systems require storage space for video footage and often require human intervention to operate. As a result, one of the main objectives of our project is to develop a system that eliminates the need for human intervention by incorporating artificial intelligence. Moreover, we aim to create a security system that relies on wireless communication. Introducing wireless communication to monitoring systems can bring numerous advantages such as improved flexibility, scalability, range, and the ability to monitor in real-time.

In the next chapter, we will delve into the various methods of artificial intelligence applicable in the security field, including machine learning and deep learning. We will also discuss their growing importance and how they are being used to enhance the accuracy and efficiency of security systems. Additionally, we will explore the specific algorithms and models being used, as well as their benefits and limitations.

Our plan is to apply these methods in an embedded system capable of operating in real-time using a powerful processing card like Nvidia Jetson Nano. The ultimate goal of our project is to merge these two technologies, wireless communication and artificial intelligence, to create a smart and efficient monitoring system that provides enhanced security measures.

I.7. References cited in Chapter I

[1] W. Ismail, S. Abdul Shukor, H. Hashim, L. A. Mutalib, and A. S. Baharuddin, "THE REALITY ON APPLICATION AND CHALLENGES OF CLOSED-CIRCUIT TELEVISION (CCTV) IMAGES AS EVIDENCE IN SHARIAH CRIMINAL CASES IN MALAYSIA," *Humanities & Social Sciences Reviews,* 2019.

[2] K. K. Lee, "High definition digital cctv system," 2011.

[3] M. Gill and K. Loveday, "What do offenders think about CCTV?," *Crime prevention and community safety,* vol. 5, pp. 17-25, 2003.

[4] H. Turtiainen, A. Costin, T. Lahtinen, L. Sintonen, and T. Hamalainen, "Towards large-scale, automated, accurate detection of CCTV camera objects using computer vision. Applications and implications for privacy, safety, and cybersecurity.(Preprint)," *arXiv preprint arXiv:2006.03870,* 2020.

[5] A. Dimou, P. Medentzidou, F. A. Garcia, and P. Daras, "Multi-target detection in CCTV footage for tracking applications using deep learning techniques," in *2016 IEEE international conference on image processing (ICIP)*, 2016, pp. 928-932.

[6] D. Serebrennikov and D. Skougarevskiy, "A tale of four cities: Exploring environmental characteristics of CCTV equipment placement," *Available at SSRN,* 2022.

[7] D.-H. Han, "Design and Characteristics of 6-60 Lens for CCTV," *Journal of Convergence Society for SMB,* vol. 6, pp. 85-91, 2016.

[8] J. Myrans, R. Everson, and Z. Kapelan, "Automated detection of fault types in CCTV sewer surveys," *Journal of Hydroinformatics,* vol. 21, pp. 153-163, 2019.

[9]https://www.fudechards.pw/products.aspx?cname=analog+cctv+camera+black+screen&cid=63.

[10] https://platinumcctv.com/analog-security-cameras.html.

[11] https://kintronics.com/resources/ip-camera-system-resources/.

[12] https://www.amazon.com/ZOSI-Wireless-System-1080p-Weatherproof Detection/dp/B00MP56XBS.

[13] https://www.spycameracctv.com/collections/wireless-cctv.

[14] Kyung Kook Lee, "High definition digital cctv system," 2011.

[15] H. U. Keval, "CCTV Control Room Collaboration and Communication: Does it Work?," 2006.

[16] N. Zurawski and S. Czerwinski, "Crime, Maps and Meaning: Views from a Survey on Safety and CCTV in Germany," *surveillance and society,* vol. 5, 2002.

[17] https://www.cctvsg.net/how-do-cctvs-work/.
[18] N. Haering, P. L. Venetianer, and A. Lipton, "The evolution of video surveillance: an overview," *Machine Vision and Applications,* vol. 19, pp. 279-290, 2008/10/01 2008.
[19] A. J. Lipton, C. H. Heartwell, N. Haering, and D. Madden, "Automated video protection, monitoring & detection," *IEEE Aerospace and Electronic Systems Magazine,* vol. 18, pp. 3-18, 2003.
[20] J.-Y. Dufour, *Intelligent video surveillance systems*: John Wiley & Sons, 2012.
[21] A. Prati, R. Vezzani, M. Fornaciari, and R. Cucchiara, "Intelligent Video Surveillance as a Service," in *Intelligent Multimedia Surveillance: Current Trends and Research*, P. K. Atrey, M. S. Kankanhalli, and A. Cavallaro, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1-16.
[22] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, "Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing," *IEEE Transactions on Industrial Informatics,* pp. 1-1, 2019.
[23] D. R. Patrikar and M. R. Parate, "Anomaly detection using edge computing in video surveillance system: review," *International Journal of Multimedia Information Retrieval,* vol. 11, pp. 85-110, 2022/06/01 2022.
[24] R. Armitage, G. Smyth, and K. Pease, "Burnley CCTV evaluation," *Surveillance of public space: CCTV, street lighting and crime prevention,* vol. 10, pp. 225-50, 1999.
[25] https://platinumcctv.com/analog-security-cameras.html.,
[26] R. Oppliger, *Security technologies for the world wide web*: Artech House, 2003.
[27] H. Venter and J. H. Eloff, "A taxonomy for information security technologies," *Computers & Security,* vol. 22, pp. 299-307, 2003.
[28] J. P. Anderson, "Computer security technology planning study," ANDERSON (JAMES P) AND CO FORT WASHINGTON PA FORT WASHINGTON1972.
[29] https://www.elprocus.com/understanding-about-types-of-access-control-systems/.
[30] https://www.businessnewsdaily.com/9067-choosing-a-surveillance-system.html.
[31] E. Conrad, S. Misenar, and J. Feldman, "Chapter 7 - Domain 7: Security operations," in *Eleventh Hour CISSP® (Third Edition)*, E. Conrad, S. Misenar, and J. Feldman, Eds., ed: Syngress, 2017, pp. 145-183.
[32] https://www.fireengineers.in/services/total-flooding-system/.
[33] https://instel.es/en/a-successful-perimeter-protection-network-cameras-analytical-video/.
[34] https://taisei.com.vn/.
[35] https://www.sprintzeal.com/.
[36] https://www.electronicshub.org/wireless-communication-introduction-types-applications/.
[37] T. Jiang, L. Zhao, H. Liu, D. Deng, A. Luo, Z. Wei, *et al.*, "Performance Improvement for Mixed RF–FSO Communication System by Adopting Hybrid Subcarrier Intensity Modulation," *Applied Sciences,* vol. 9, p. 3724, 09/06 2019.
[38] https://data-flair.training/.
[39] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz, "Wireless communication technologies for ITS applications [Topics in Automotive Networking]," *IEEE Communications Magazine,* vol. 48, pp. 156-162, 2010.
[40] X. Feng, F. Yan, and X. Liu, "Study of wireless communication technologies on Internet of Things for precision agriculture," *Wireless Personal Communications,* vol. 108, pp. 1785-1802, 2019.
[41] F. Montori, L. Bedogni, M. Di Felice, and L. Bononi, "Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues," *Pervasive and Mobile Computing,* vol. 50, pp. 56-81, 2018.

[42] A. Usman and S. H. Shami, "Evolution of communication technologies for smart grid applications," *Renewable and Sustainable Energy Reviews,* vol. 19, pp. 191-199, 2013.

[43] T. M. Saad, "Détecteur et brouilleur de téléphones mobiles", Mémoire : Ingénieur CNAM En Electroniques," Institut des sciences appliquées et economiques, Université Libabaise, 2010.

[44] https://www.cdt21.com/design_guide/simplex-duplex-and-communication-protocol/.

[45] A. Ens, L. M. Reindl, T. Janson, and C. Schindelhauer, "Low-power simplex ultrasound communication for indoor localization," in *2014 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 731-735.

[46] J. Lee and T. Q. Quek, "Hybrid full-/half-duplex system analysis in heterogeneous wireless networks," *IEEE transactions on wireless communications,* vol. 14, pp. 2883-2895, 2015.

[47] M. Duarte and A. Sabharwal, "Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, 2010, pp. 1558-1562.

[48] https://www.geeksforgeeks.org/how-gsm-works/.

[49] L. Chamek, ""Localisation des mobiles par une stratégie de prédiction," Localisation des mobiles par une stratégie de prédiction, Mémoire de Magister, Option: Specification logiciels et traitement de l'information," Université M'hamed Bougara, Boumerdes, 2011.

[50] R. CODE, "SAO/NASA ADS (null) Abstract Service."

Chapitre II

Artificial Intelligence

II.1. Introduction

Today, the application of artificial intelligence includes several major scientific areas, such as natural language processing (NLP), computer vision, robotics, self-driving cars, detection, and healthcare. Machine Learning and Deep Learning are two most important concepts that make artificial intelligence possible.

Below we give brief definitions of terms artificial intelligence, machine learning and deep learning, the Figure II.1 show the relationship between them:

Artificial intelligence (AI): Refers to a field of computer science dedicated to creating systems that can perform tasks that typically require human intelligence. It can be loosely interpreted as incorporation human intelligence into machines [1].

Machine learning (ML): Is a branch of AI, which includes all the approaches that allow machines to learn from data without being explicitly programmed. The intention of ML is to train machines based on the provided data and algorithms [1].

Deep learning (DL): Is a subset of machine learning that involves neural networks with a large number of layers and parameters. Most deep learning methods use neural network architectures [2].

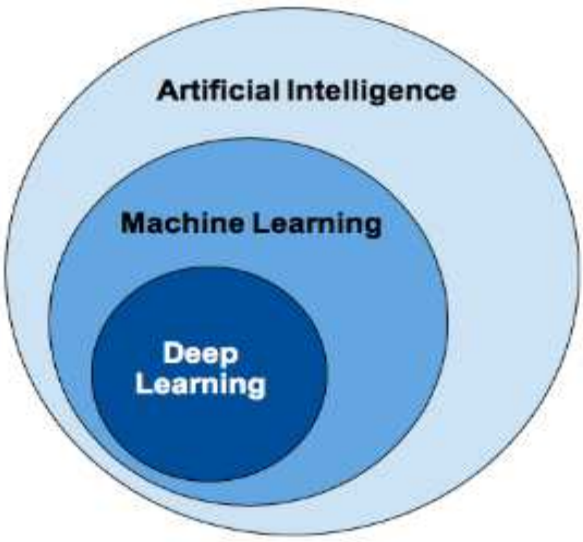


Figure II. 1. Relationship between Artificial intelligence, Machine Learning and Deep Learning.

The objective of this chapter is to provide an overview of the fundamental concepts of various machine learning, and deep learning methods that will be utilized in our project for real-time applications, specifically in the context of a smart monitoring system that employs deep learning techniques.

II.2. Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) is a very common word that may imply many different things. It may indicate any form of technology that includes some intelligent aspects rather than pinpoint a specific technology field. In contrast, Machine Learning refers to a specific field. In other words, we use Machine Learning to indicate a specific technological group of Artificial Intelligence. Machine Learning itself includes many technologies as well. One of them is Deep Learning, which is the subject of this work.

Machine learning (ML) is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of the so-called big data. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and computational biology to biomedical and medical applications [3-5].

Machine learning relies on different algorithms to solve data problems. Data scientists like to point out that there is no single algorithm that is best for solving a problem. The algorithm used depends on the problem to be solved, the number of variables, the model that best fits it, etc. As show in Figure II.2, some of the more commonly used algorithms in machine learning [6].

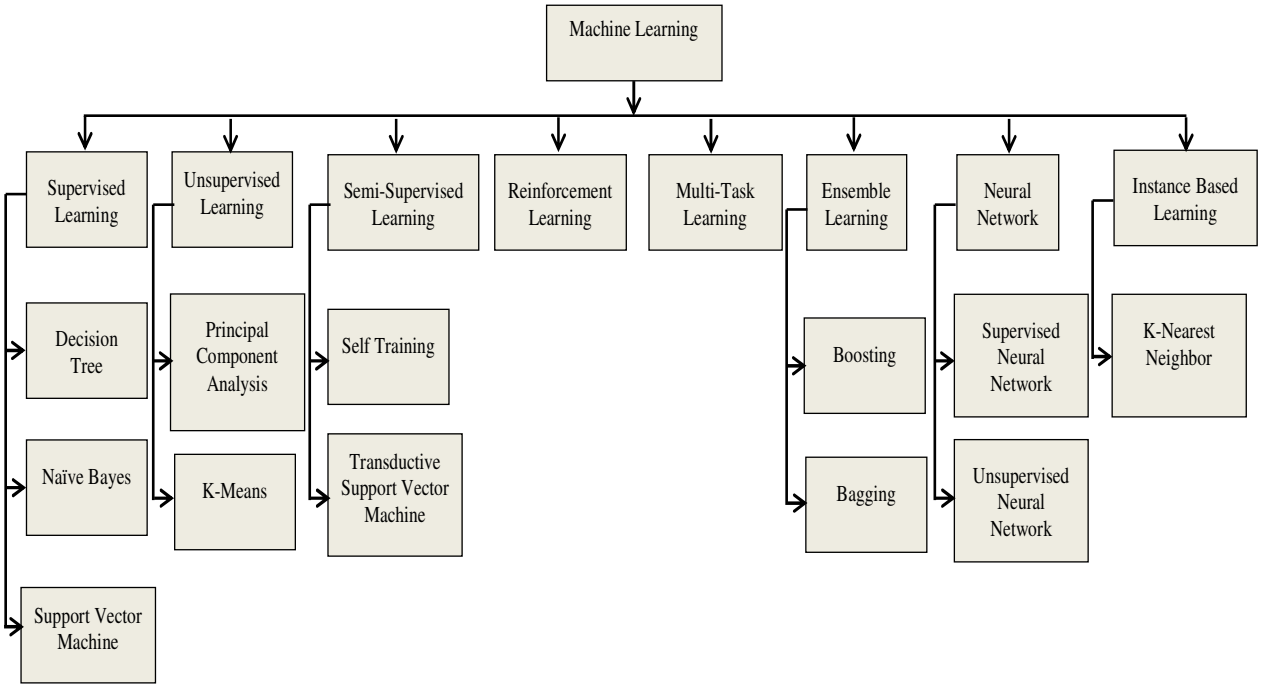


Figure II. 2. Types of algorithms in machine learning.

In scientific literature, many different types of machine learning techniques have been developed to solve problems in various fields. These Machine Learning techniques can be classified into two types based on the training method: Supervised learning and unsupervised learning, as discussed in the following subsections (II.2.1 and II.2.2).

In sections II.3 and II.4, we introduce two popular types of machine learning algorithms: artificial neural networks (ANNs) and support vector machines (SVMs). ANNs and SVMs are widely used in various applications due to their effectiveness in handling complex data and making accurate predictions.

II.2.1. Supervised Learning

Supervised learning is the Machine Learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification. The workflow of supervised machine learning algorithms is given in Figure II.3 [7, 8].

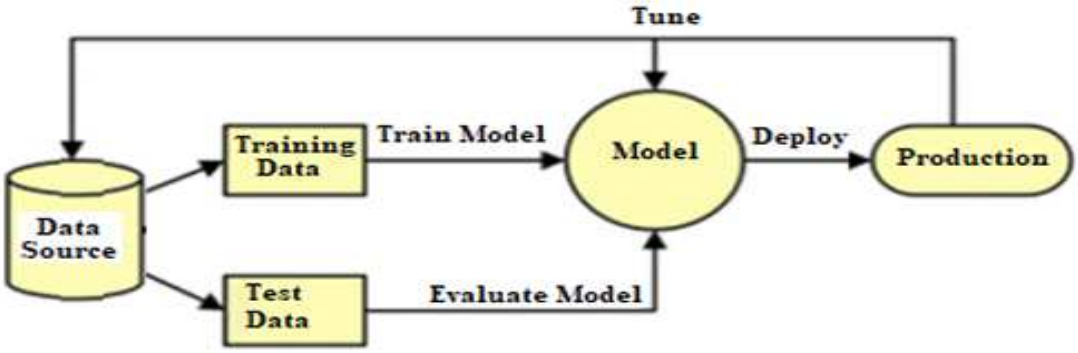


Figure II. 3. Supervised learning workflow.

II.2.2. Unsupervised Learning

These are referred to as unsupervised learning because, in contrast to the supervised learning described above, there are no correct answers and there are no teachers. The algorithms are left to find and display the intriguing structure in the data on their own. Few features are learned from the data by the unsupervised learning algorithms. When new data is introduced, it recognizes the class of the data using the previously learnt features. It is mostly utilized for feature reduction and grouping [9].

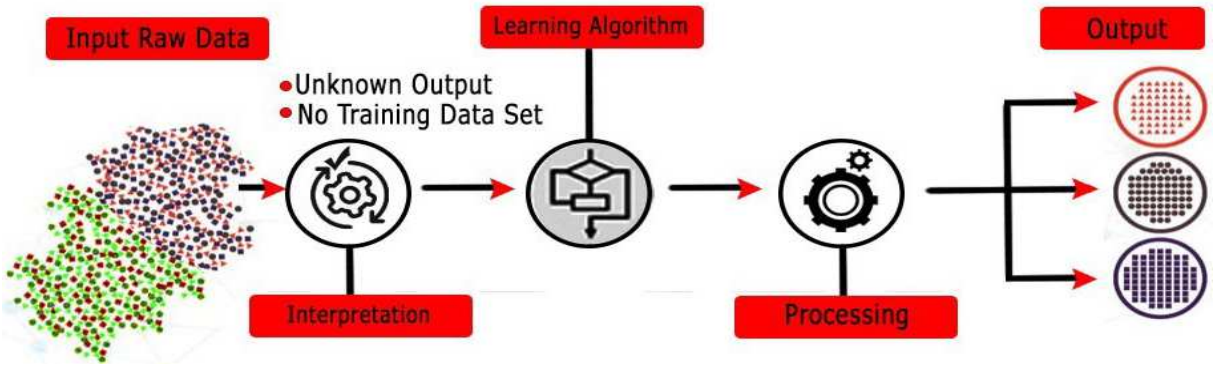


Figure II. 4. Unsupervised Learning.

II.3. Artificial Neural Networks

Artificial neural network (ANN) is a category of computing systems that can exhibit generalization for a given task beyond the training data [10]. The concept of ANN is basically introduced from the subject of biology where neural network plays an important and key role in human body. In human body work is done with the help of neural network. Neural Network is just a web of inter connected neurons which are millions and millions in number. With the help of these interconnected neurons all the parallel processing is done in human body [11].

A biological neuron is illustrated in Figure II.5 (a), similar to the biological neurons, the artificial ones (see Figure II.5 (b)) have input and output, bias or threshold, connections where each connection has a weight to represent the connection strength between units, and also an activation function [12].

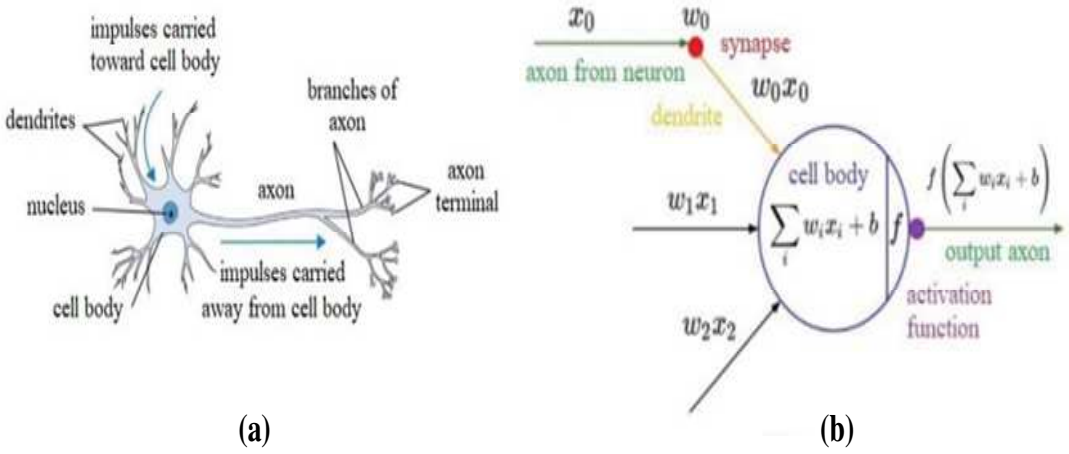


Figure II. 5. (a) Biological neuron, and (b) Artificial neuron.

Mathematically, the neural network is expressed as the inputs weighted sum, by multiplying each input with their respective weight and then sum all the results:

$$\hat{y} = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{II.1}$$

where n is the number of inputs (features) and f is the activation function, the output layer's activation is denoted as $\hat{y}$. The process of computing the output $\hat{y}$ from the input x is called forward propagation since the information propagates from the input layer to the output layer.

II.3.1. Multilayer Perceptrons Network MLP

Multilayer perceptrons (MLPs) also called multilayer feed-forward neural networks, are very popular and are used more than other neural network types for a wide variety of problems. MLPs are based on a supervised procedure, i.e., the network builds a model based on examples in data with known outputs. An MLP has to extract this relation solely from the presented examples, which together are assumed to implicitly contain the necessary information for this relation [13].

II.3.1.1. MLP Structure

The MLP network architecture consists of:

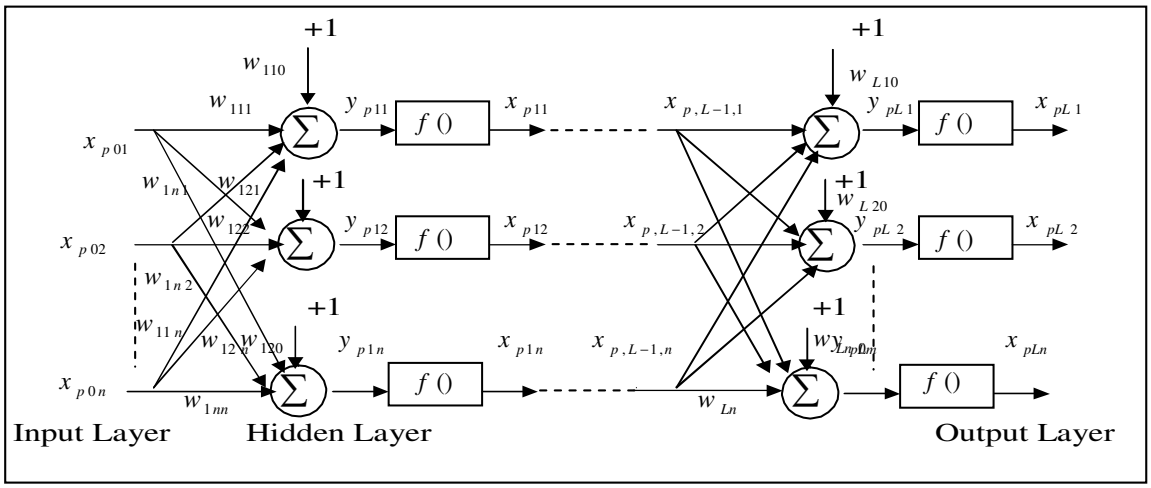


Figure II. 6. Fully Connected Feed Forward Multilayer Perceptron.

- *An input layer:* there are the values provided to the neuron for computation.
- *A hidden layer or (layers):* which contains the neurons (nodes) that contain an activation function that determines the behavior of the node. The inputs pass to the first hidden layer via the input layer, where the number of nodes contained in the input layer must be equal to the number of input features. Within the first hidden layer, the sum of each of the inputs with a given weight is calculated with the addition of the bias value according to the following equation [14]:

$$y_{plk} = \sum_{l=1}^{L} w_{plk} x_{p,l-1,k} + b_{p,l-1,k} \quad \text{(II.2)}$$

Thus the result of the neuron can be represented as:

$$x_{plk} = f\left(y_{plk}\right) \quad \text{(II.3)}$$

The MLP usually uses a *gradient descent method* called *backpropagation* (BP) of errors as a learning algorithm. The BP algorithm for gradient of error is a way of calculating how an entity's derivatives change with respect to its input set. It is used to solve the problem of computing the output errors of neurons in hidden layers. This algorithm adjusts the weights of a network with a fixed architecture determined by the user, with the goal of minimizing the squared error between the computed outputs and the desired outputs.

The error on the output node is:

$$\delta_{pk} = O_{PK} - x_{plk} \quad \text{(II.4)}$$

Consequently, the total error (for all nodes) is:

$$E_P = \frac{1}{2}\sum_{k=1}^{m} \delta^2{}_{p,k} = \frac{1}{2}\sum_{k=1}^{m} (O_{p,k} - x_{p,l,k})^2 \quad \text{(II.5)}$$

To minimize *Ep* (quadratic error), we calculate its gradient with respect to each weight *w*, and then adjust the weights using the BP algorithm.

II.3.1.2. Activation Functions

The choosing of the activation functions is often empirical, depending on the specific dataset. The most commonly used and widely mentioned functions in scientific literature are listed below:

The three most frequently used functions are: (a) "Threshold" (also known as "hard limit"), (b) "Linear" and (c) "Sigmoid" as illustrated below:

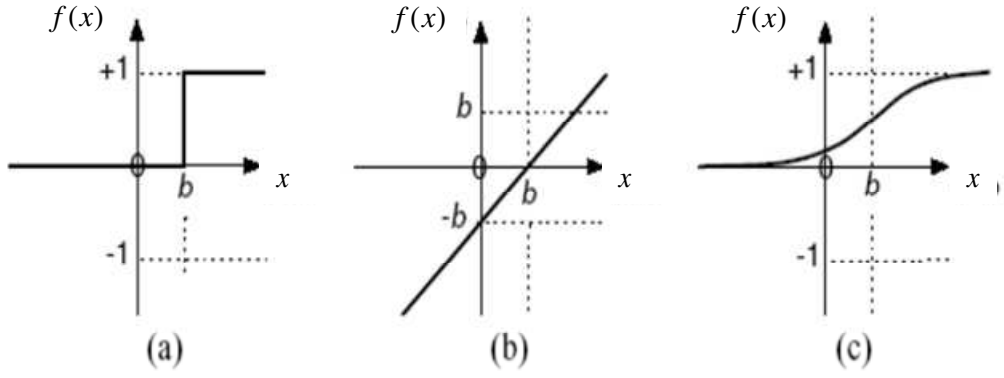


Figure II.7. Three activation functions: (a) Threshold, (b) Linear, and (c) Sigmoid.

II.3.2. Radial Basis Function Neural Networks

Radial basis function neural networks approaches, developed by Broomhead and Lowe in 1988; they are feed-forward networks which are trained by a supervised algorithm. They have been broadly used in classification and interpolation regression tasks. Comparing to other neural networks, the RBFNNs are faster in their training phase and provide a better approximation due to their simpler network architecture [15].

II.3.2.1. Network Architecture

The standard radial basis function (RBF) neural network consists of three layers: an input layer, a hidden layer, and an output layer. Figure II.8 show a schematic representation of the RBF network. The number of the nodes in the input and output layers is decided by the research objects [16]. The input layer can have more than one predictor variable where each variable is associated with one independent neuron. The output of the input layer neurons, then feed the values to each of the neurons in the hidden layer. The hidden layer can have multiple numbers of neurons [17].

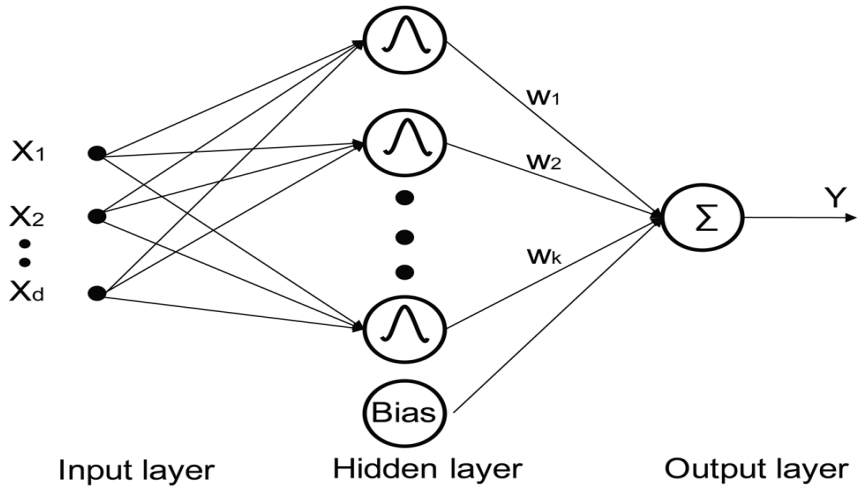


Figure II. 8. Architecture of an RBF neural network.

Each node in the hidden layer determines a level of activation of the receptive field (radial basis function) $\Phi(x)$ given some input x. The j^{th} output $y_j(x)$ is a weighted of linear combination of the activation levels of the receptive fields:

$$y_j(x) = \sum_{i=1}^{c} w_{ij} \Phi_i(x) \tag{II.6}$$

The most commonly used radial basis function is a Gaussian kernel function, also called robustness. Radial basis kernel function, the typical form is:

$$\Phi(x) = \exp\left(-\frac{\|x - v_i\|}{2\sigma^2}\right) \quad \text{(II.7)}$$

where x is the n-dimensional input vector $[x_1, x_2, ..., x_n]^T$, and $[v_1, v_2, ..., v_n]^T$ the center of i^{th} the basis function $\Phi_i(x)$ while c is the number of the nodes in the hidden layer [18].

RBF networks are used as approximation tools in various cases, such as solutions to differential equations, digital communications, physics, chemistry, economics, network security, etc [19].

II.4. Support Vector Machine SVM

Support vector machine (SVM) was first heard in 1992, introduced by Boser, Guyon, and Vapnik. Support vector machines (SVMs) are a set of related supervised learning methods [20] with associated learning algorithms that analyze data and recognize patterns used for classification and regression analysis [21], these models are based on discriminating between two classes of observations by a linear decision surface (hyperplane): maximizing the distance between the hyperplane and the individual observations. If the classes are not separable by a linear surface, a non-linear transformation can be obtained through mapping the data on a different dimension space (feature space), usually a much higher dimensional [22].

II.4.1. Classification Mode

The SVM analyzed two kinds of data, linearly and non-linearly separable data:

II.4.1.1. Linear Classification

For each training sample (y_i,x_i), y_i represents its class, and x_i represents its input vector defined on a d-dimensional space. Suppose the training samples: $\{(y_1, x_1), (y_2, x_2), ...(y_n, x_n)\}$ $x_i \in R^d$, $y_i \in \{-1, +1\}$ [23], suppose we have some hyperplane which separates the positive from the negative examples "separating hyperplane". The points x which lie on the hyperplane satisfy:

$$(x \bullet w) + b = 0 \quad \text{(II.8)}$$

where w is normal to the hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyperplane to the origin, and $\|w\|$ is the Euclidean norm of w.

$$\begin{cases} (x_i \bullet w) + b \geq +1 & for \quad y_i = +1 \\ (x_i \bullet w) + b \leq -1 & for \quad y_i = -1 \end{cases} \quad \text{(II.9)}$$

These can be combined into one set of inequalities [24]:

$$y_i(x_i \bullet w) + b - 1 \geq 0 \quad \text{(II.10)}$$

Among the separating hyperplanes, the one for which the distance to the closest point is maximal, called optimal separating hyperplane (OSH) [25].

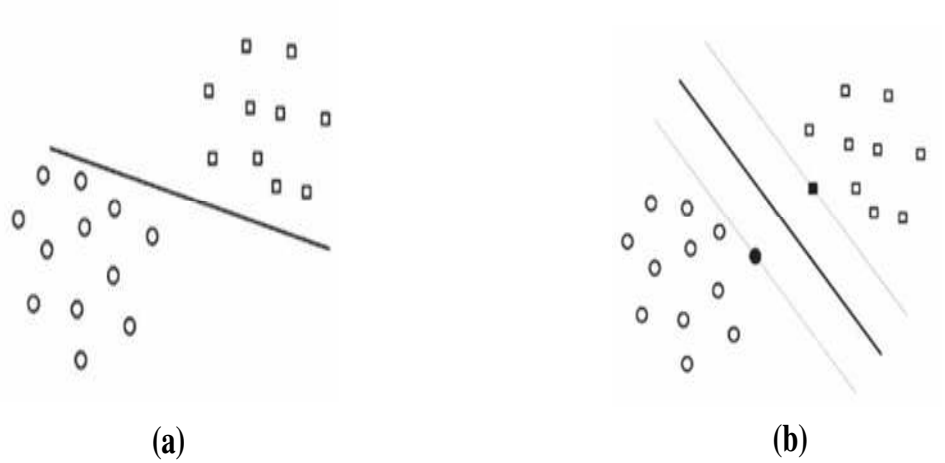


Figure II. 9. (a) Separating Hyperplane, (b) Optimal Separating Hyperplane.

II.4.1.2. Non Linear Classification

For non-linear separable patterns, the data may be mapped into a high-dimensional space through some nonlinear mapping which has the effect of spreading the distribution of the data in a way that facilitates the fitting of a linear hyperplane (see Figures II.10 and II.11). The kernel function enables us to implicitly work in a higher dimensional feature space. An input data point x can be represented as $\varphi(x)$ in the high-dimensional space H. The expensive computation of $(\varphi(x),\varphi(x_i))$ in a high-dimensional space is reduced by using a positive definite *kernel* such that $(\varphi(x),\varphi(x_i))=k(x,x_i)$.

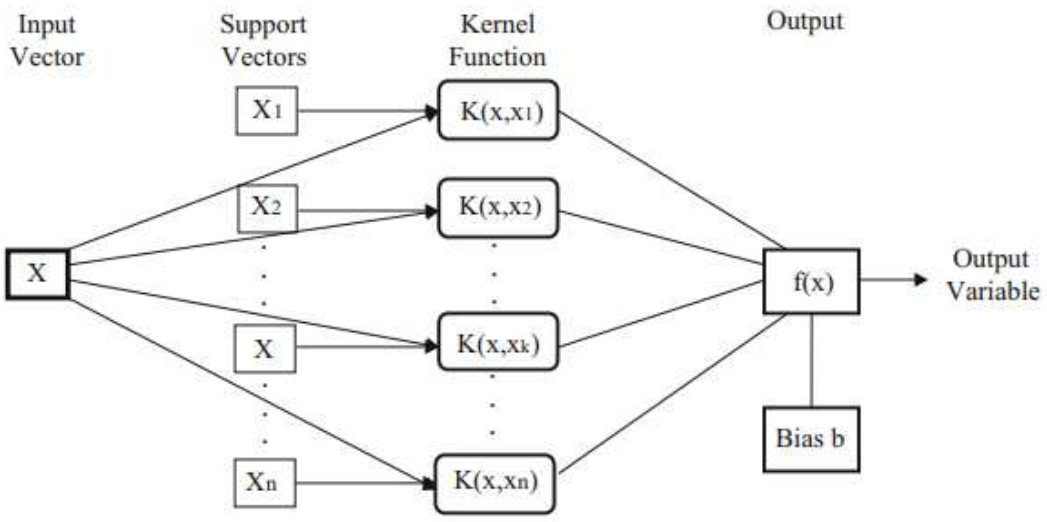


Figure II. 10. A schematic structure of SVM model.

Leading to decision functions of the form:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{r} \alpha_i y_i k(x, x_i) + b\right) \quad \text{(II.11)}$$

where α_i is a Lagrange multiplier [26].

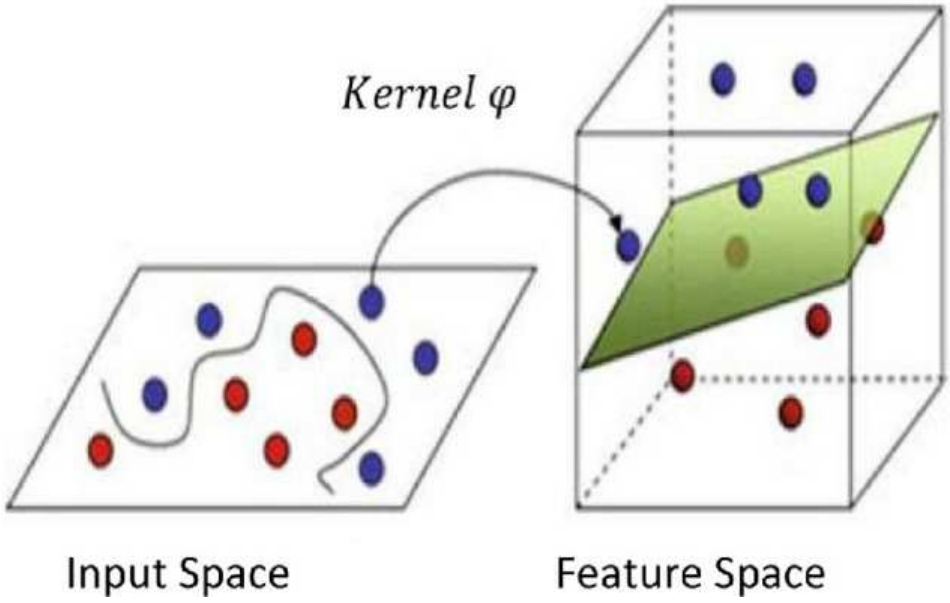


Figure II. 11. Feature space is related to input space via a nonlinear map φ, causing the decision surface to be nonlinear in the input space.

To choose the kernel K: the most commonly used kernels in various applications are [27]:

- **Gaussian radial basis function:**

$$K(x_i, x) = \exp\left(-\gamma \left\| x_i - x \right\|^2\right) \quad \text{(II.12)}$$

γ : Parameter is inversely proportional to the width of the Gaussian, where $\gamma = \frac{1}{2.\sigma^2}$.

- **Polynomial kernel:**

$$K(x_i, x) = \left(x_i^T .x + 1\right)^d \quad \text{(II.13)}$$

where *d* is the degree of the polynomial.

- **Linear kernel:**

$$K(x_i, x) = \left(x_i^T .x\right) \quad \text{(II.14)}$$

II.5. Deep Learning

Deep learning is indeed an extension of neural networks, and most of what you have read previously is applicable. Knowing that Deep Learning is just the use of a deeper (more hidden

layers) neural network, the questions may be asked are: "*What makes Deep Learning so attractive*? *Has anyone ever thought of making the neural network's layers even deeper*?" In order to answer these questions, we need to look into the history of the neural network.

It did not take very long for the single-layer neural network, the first generation of the neural network, to reveal its fundamental limitations when solving the practical problems that Machine Learning faced. The researchers already knew that the multi-layer neural network would be the next breakthrough. However, it took approximately 30 years until another layer was added to the single-layer neural network. It may not be easy to understand why it took so long for just one additional layer. It was because the proper learning rule for the multi-layer neural network was not found. Since the training is the only way for the neural network to store the information, the untrainable neural network is useless [28].

The reason the multi-layer neural network took 30 years to solve the problems of the single-layer neural network was the lack of the learning rule, which was eventually solved by the back-propagation algorithm. In contrast, the reason another 20 years passed until the introduction of deep neural network-based Deep Learning was the poor performance. The backpropagation training with the additional hidden layers often resulted in poorer performance. Deep Learning provided a solution to this problem [28].

Deep learning is a machine learning method that uses multiple levels of nonlinear processing units, where each layer uses the previous layers output as input. This includes neural networks with multiple hidden layers between the input and output layers. As each layer transforms the data, it would ideally mean that each layer successively represents a higher level of abstraction about the data [29].

Deep learning models usually adopt hierarchical structures to connect their layers. The output of a lower layer can be regarded as the input of a higher layer via simple linear or nonlinear calculations. These models can transform low-level features of the data into high-level abstract features [30].

The most commonly used type of deep learning architecture is Convolutional Neural Networks, which are particularly well-suited for image and video processing tasks.

II.5.1. Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which is mostly applied to computer vision tasks or to analyze images and videos in a wide range of applications, for example, image classification and object detection. The CNNs are not only designed for processing the 2 Dimensional (2D) data like images but also CNNs can be

modified into 1D networks to process sequential time-dependent data, as well as into 3D networks to be applied for video processing [31]. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. ConvNets have the ability to learn these filters/characteristics [32].

II.5.1.1. CNN Architecture

CNN is a mathematical construct that is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers [33]. The two first, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification. A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation [34].

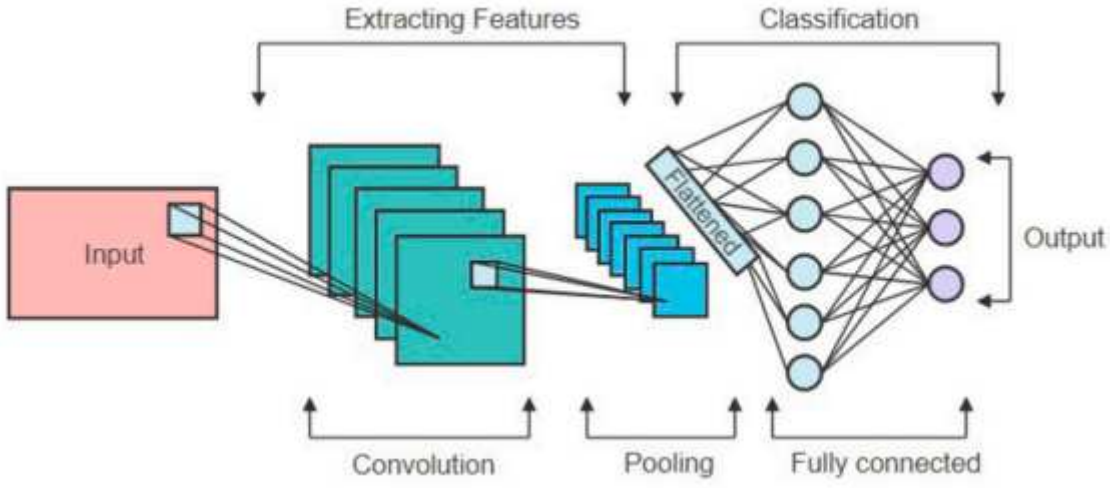


Figure II. 12. Convolutional Neural Network.

II.5.1.1.1. Convolutional Layer

The first layer of each CNN used is 'input layer' which takes images, resize them for passing onto further layers for feature extraction [35].

This layer is the primary unit of CNN in which most of computations are performed. These layers consist of arrangement of neurons along with set of feature maps. These layers have filters (kernels) that are used to convolve with features for producing a separate 2-D activation map as shown in Figure II.13 [36].

Convolutional layers are also able to significantly reduce the complexity of the model through the optimization of its output. These are optimized through three hyper parameters, the depth, the stride and setting zero-padding [13].

- Stride and Padding values are two important parameters for a filter for deciding the size of the output image.
- Stride value says the number of columns a filter should shift after a convolution operation [37].
- Depth of the output volume produced by the convolutional layers can be manually set through the number of neurons within the layer to the same region of the input.
- Zero-padding is the simple process of padding the border of the input, and is an effective method to give further control as to the dimensionality of the output volumes [38].

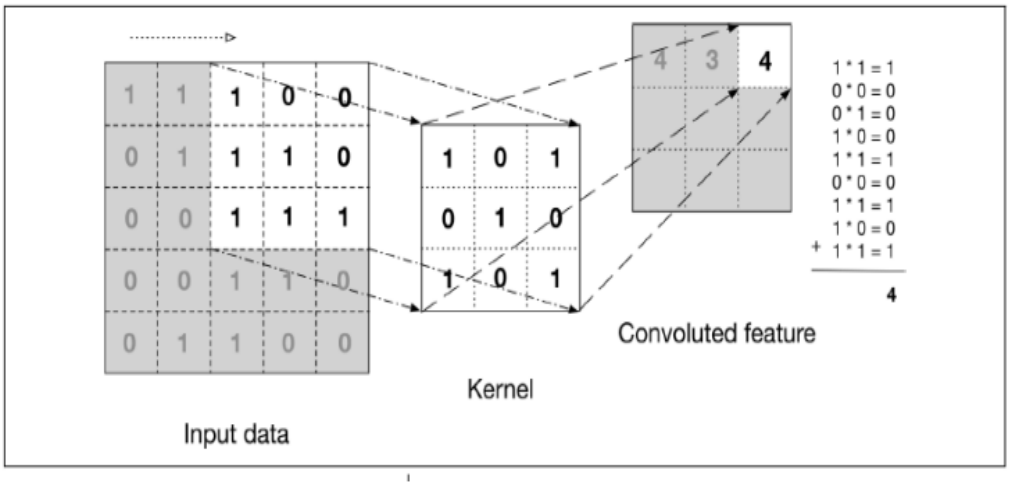


Figure II. 13. The convolution operation.

II.5.1.1.2. Pooling Layer

Pooling layers used to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model. It makes the features strong against noise and distortion. There are two methods to do pooling: max pooling and average pooling. In two cases, the input is divided into two-dimensional spaces [27]. As shown in Figure II.14, an example of pooling is depicted.

- **Max pooling:** Takes the maximum value among all values in the collection window and it is the most common type.
- **Average pooling:** Takes the average of values in the collection window [39].

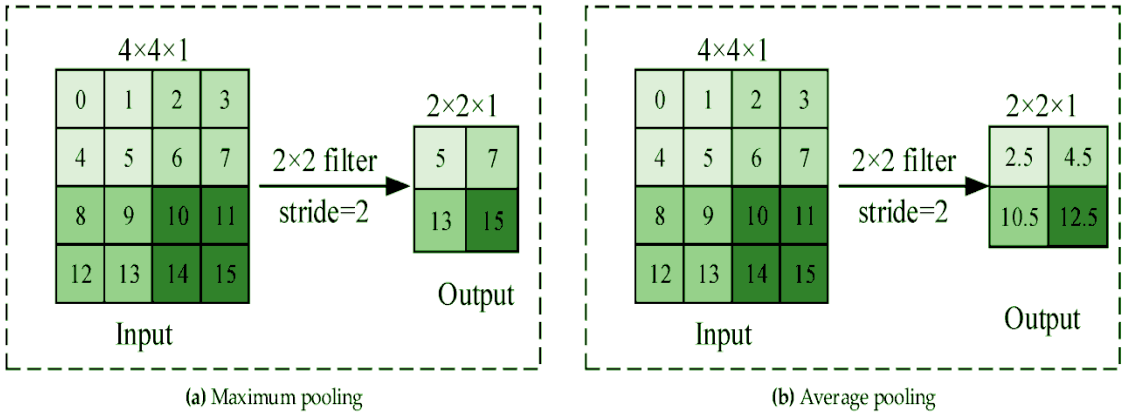


Figure II. 14. The process of pooling operation.

II.5.1.1.3. Nonlinearity Layer (Activation Function)

The activation function plays a vital role in CNN layers. The output of the filter is provided to another mathematical function called an activation function. ReLu, which stands for rectified linear unit, is the most common activation function used in CNN feature extraction. The main motive behind using the activation function is to conclude the output of neural networks, such as yes or no. The activation function maps the output values between −1 to 1 or 0 to 1, etc. [40]. ReLU function is illustrated in Figure II.15, with its transfer function plotted above the arrow [41].

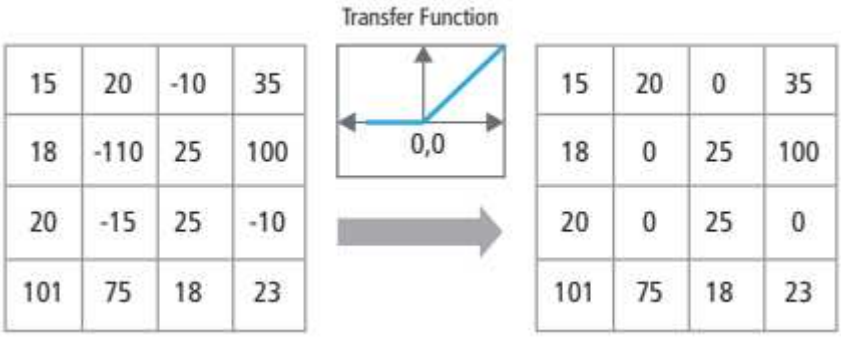


Figure II. 15. Pictorial representation of ReLU functionality.

II.5.1.1.4. Fully Connected Layer

Commonly, this layer is located at the end of each CNN architecture. Inside this layer, each neuron is connected to all neurons of the previous layer, the so-called Fully Connected (FC) approach. It is utilized as the CNN classifier. It follows the basic method of the conventional multiple-layer perceptron neural network, as it is a type of feed-forward ANN. The input of

the FC layer comes from the last pooling or convolutional layer [42]. To be inputted in the fully connected layers, the features maps are flattened into a single 1D vector [43].

II.5.1.1.5. Activation Functions

The activation function plays an important role in the training of neural networks. They provide the necessary nonlinearity of the model to be able to learn complex representations [44]. Activation functions are mathematical equations; they determine which information should be transmitted to the next neuron.

The mathematical form of Activation Function:

$$Y = \sum(weight * Input) + Bais \tag{II.15}$$

a) RELU function

ReLU is by far the most popular activation function for training deep neural networks. The reason for its popularity is that it is computationally efficient it allows the network to converge very quickly [45]. Therefore, the ReLU activation function has been developed in order to find a solution to the *vanishing gradient* problem. The ReLU activation function is a non-linear activation function that can perform the derivative operation [46]. The equation of the ReLU activation function is:

$$f(x) = \max(0, x) = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x \geq 0 \end{cases} \tag{II.16}$$

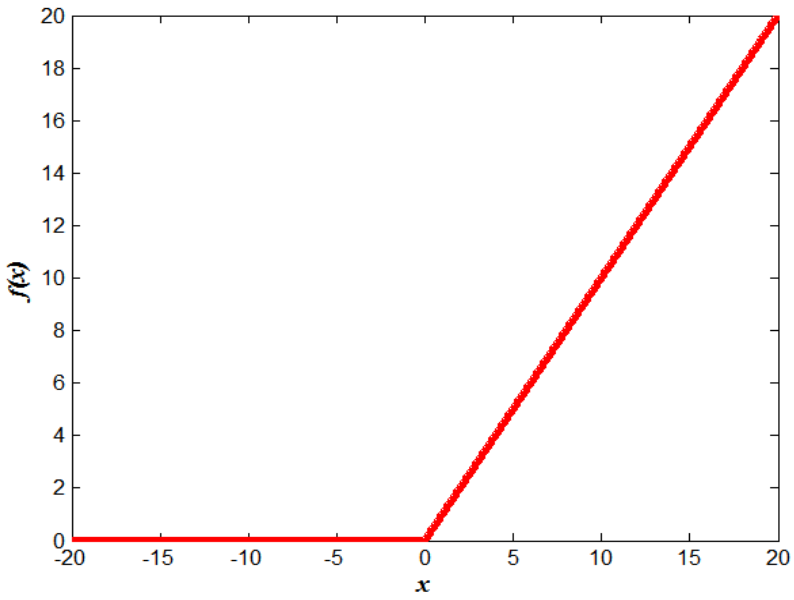


Figure II. 16. RELU function.

b) Sigmoid function

The sigmoid activation function takes real numbers as its input and binds the output in the range of [0, 1]. The curve of the sigmoid function is 'S' shaped. The mathematical representation of sigmoid is [47]:

$$f(x) = \frac{1}{1+e^{-x}} \tag{II.17}$$

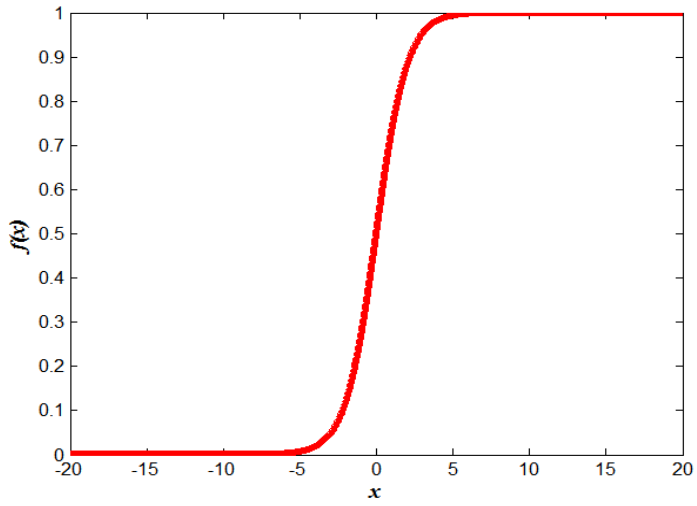


Figure II. 17. Sigmoid function.

c) Leaky Relu function

Leaky ReLU is an improvised version of ReLU function where for negative values of x, instead of defining the ReLU functions value as zero, it is defined as extremely small linear component of x. It can be expressed mathematically as [48]:

$$f(x) = \max(0.01 \bullet x, x) \tag{II.18}$$

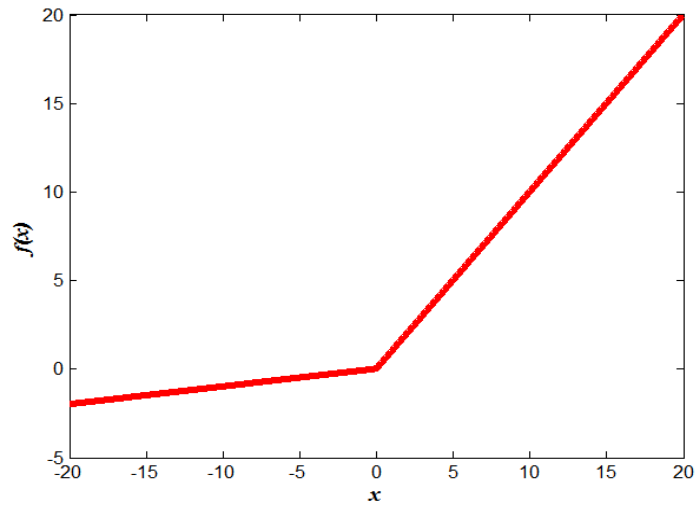


Figure II. 18. Leaky Relu function.

d) Softmax Function

The Softmax function is another types of activation function used in neural computing. It is used to compute the probability distribution from a vector of real numbers. The Softmax function produces an output which is a range of values between 0 and 1, with the sum of the probabilities been equal to 1. The Softmax function is computed using the following equation [49], Refer to Figure II.19 for the representation of this function.

$$f(x) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{II.19}$$

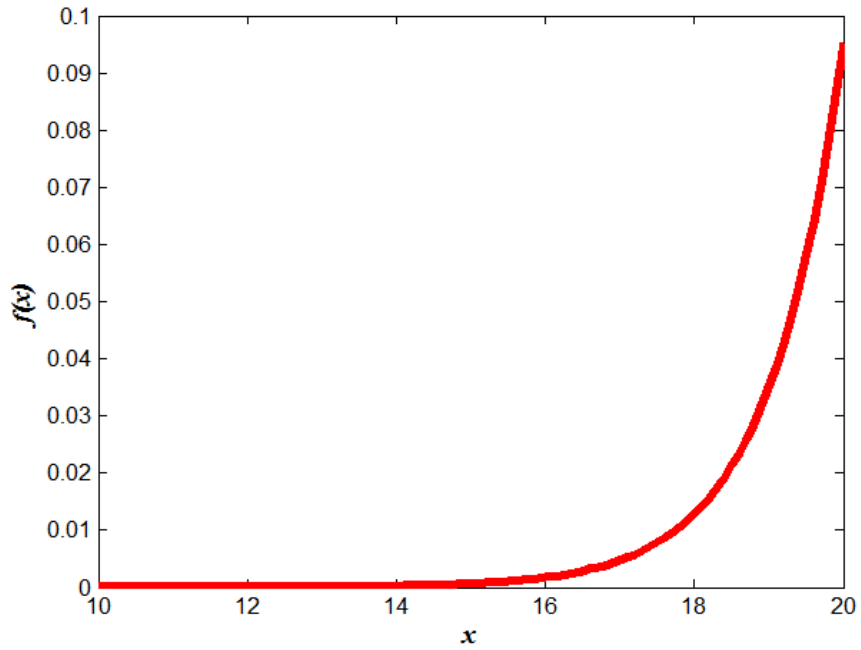


Figure II. 19. Softmax function.

e) Tanh function

From the next equation it is clear that Tanh can be considered as a scaled up version of sigmoid (see Figure II.20), outputting values in the range of -1 and 1. The problem of saturating gradients exists with this function as well but since the outputs are zero-centered, the second problem stated above is eliminated. Thus practically tanh is preferred to sigmoid [50].

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{II.20}$$

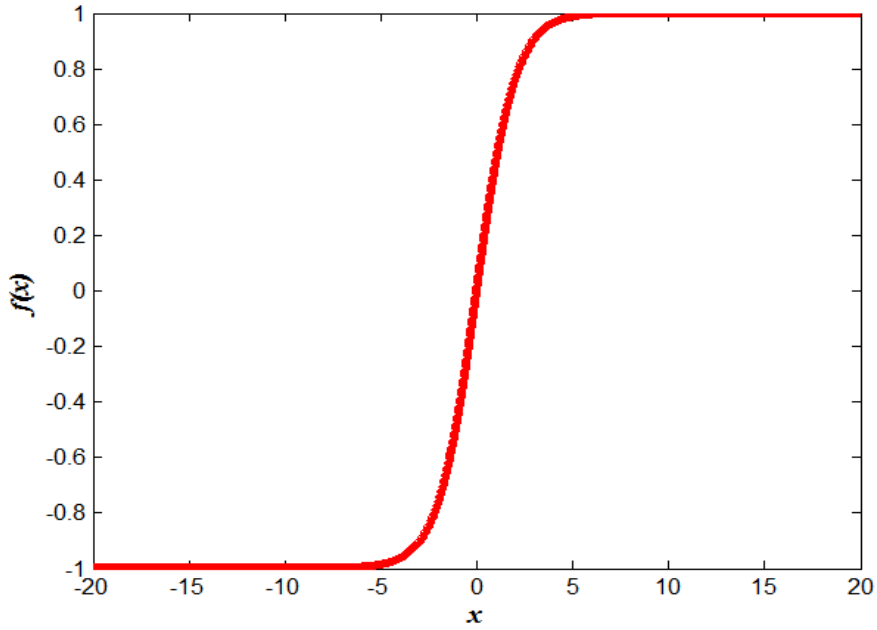


Figure II. 20. Tanh function.

II.5.2. Several Common CNN Architectures

In this section, several common transfer learning techniques are presented. Transfer learning is a machine learning approach that involves leveraging knowledge learned from one task or domain to improve the performance on a different but related task or domain. Some of the common transfer learning techniques include fine-tuning pre-trained models, using feature extraction, domain adaptation, and multi-task learning. These techniques are widely used in various applications to benefit from pre-existing knowledge and achieve better performance with limited data or resources. Examples of popular pre-trained models used in transfer learning include AlexNet, GoogLeNet, MobileNet, ResNet, VGG, etc.

II.5.2.1. AlexNet

AlexNet is a deep learning algorithm introduced by Krizhevsky et al. [51]. This deep CNN contains 25 layers consisting of 5 convolution layers, 3 max-pooling layers, 2 dropout layers, 3 FC layers, 7 relu layers, 2 normalization layers, softmax layer, input and classification (output) layer. The size of the image to be positioned in the input layer is 227×227×3 [52]. The key advantage of this network is that the training procedure is computationally efficient when compared to the other networks. However, the network is not deep enough to collect complex features from images.

In Figure II.21, we illustrate the AlexNet CNN architecture, and in Figure II.22, we provide a basic layout of the AlexNet architecture, showcasing its five convolutional layers and three fully connected layers.

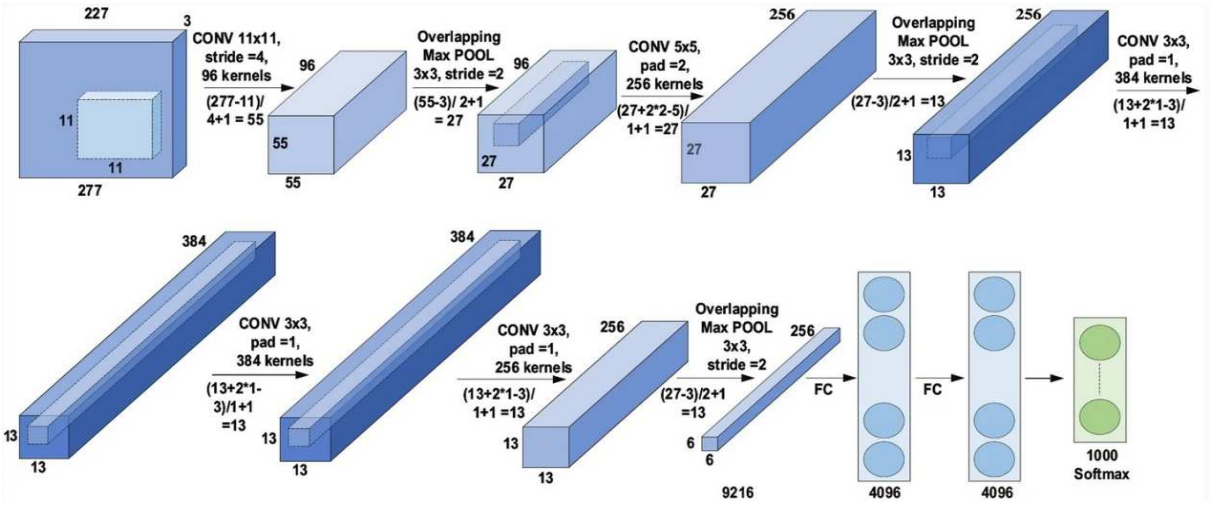


Figure II. 21. AlexNet CNN architecture.

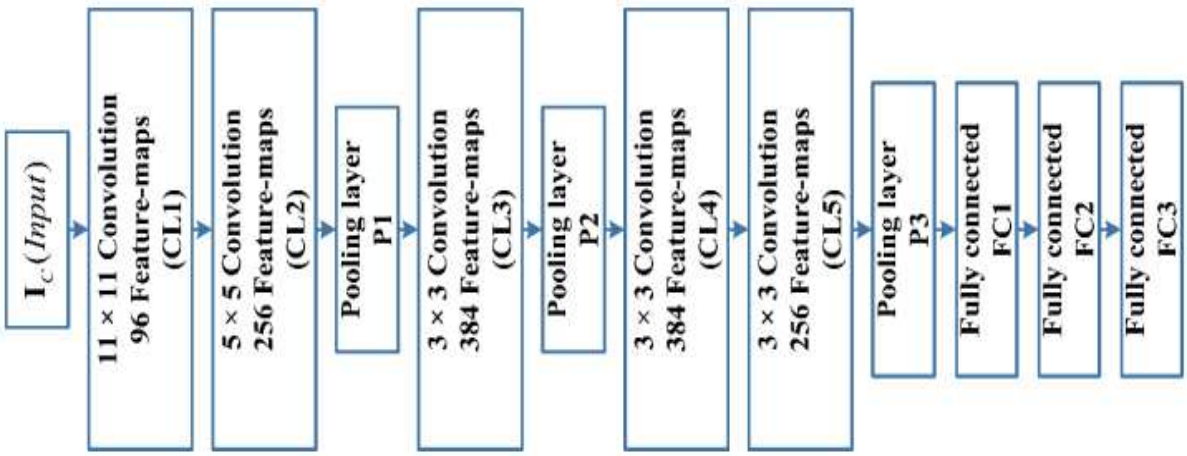


Figure II. 22. Basic layout of AlexNet architecture.

II.5.2.2. GoogLeNet

The GoogLeNet architecture was first introduced by Szegedy et al. [53]. GoogLeNet is an inception architecture that enables one to increase the width and depth of the network for an improved generalization capacity per a constant computational complexity. The GoogLeNet architecture consists of 6.8 million parameters with nine inception modules, two convolutional layers, one convolutional layer for dimension reduction, two normalization layers, four max-pooling layers, one average pooling, one fully connected layer, and a linear layer with softmax activation function in the output (see Figure II.23) [54].

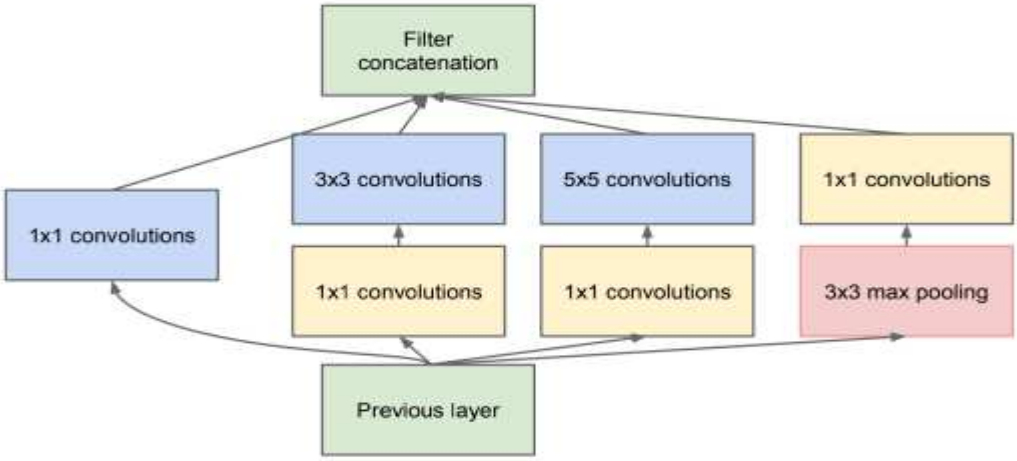


Figure II. 23. GoogLeNet inception model.

II.5.2.3. MobileNet

MobileNet was originally presented by Howard et al. [55].Its goal is to create CNN, which would be able to perform well even on computationally limited platforms such as mobile phones. MobileNet uses at its core depth wise separable convolutions, which are a form of factorised convolutions. It also introduces two global hyper parameters for balancing latency and accuracy. They are width multiplier and resolution multiplier, which help to create smaller and efficient MobileNets. Its most recent version, MobileNet-v2 which was introduced by Sandler et al. [56]. It adds inverted residual with linear bottleneck as a new layer module. Its structure includes 17 inverted residual layer modules followed by convolutional layer [57].

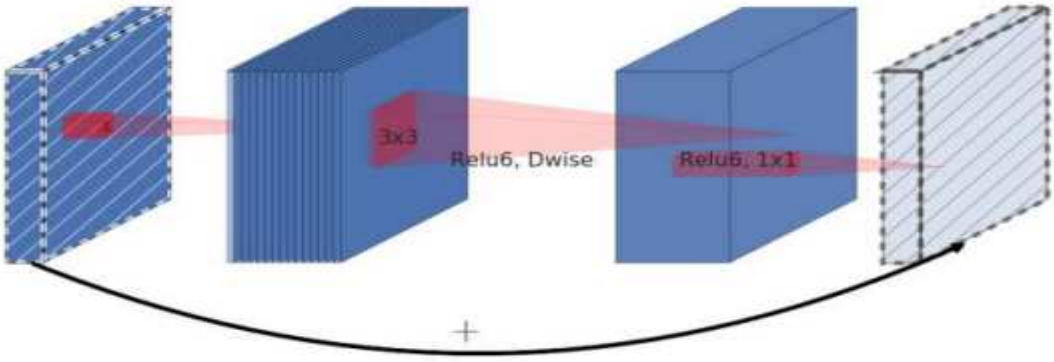
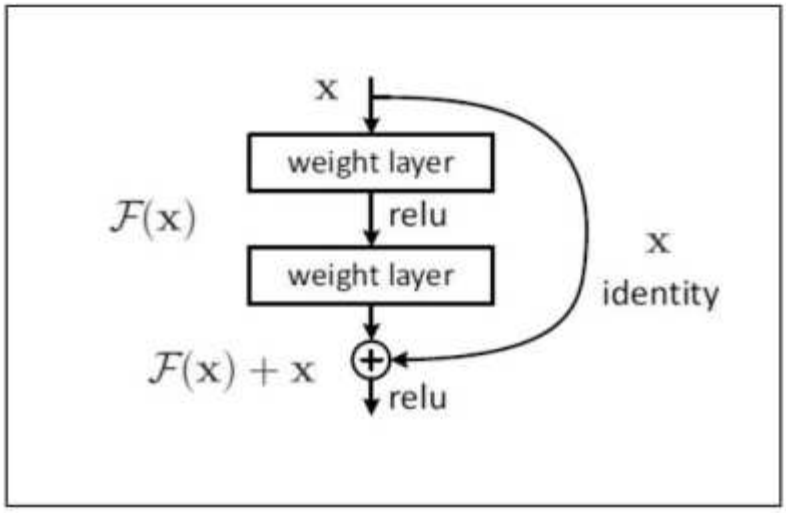


Figure II. 24. Illustration of convolutional layer used in MobileNet-v2 including depth wise separable convolutions, a shortcut connection, and linear bottleneck (hatched layers have no non-linearities).

II.5.2.4. ResNet

ResNet is a Convolutional Neural Network created in 2015 by MSRA (Microsoft Research Asia) [58].The idea of residual learning framework solves the degradation problem which is displayed when the networks depth increases. It causes lesser parameters and tackles vanishing gradient problem. The experiment shows deeper ResNets have the lower training and 21 test error. In order to solve this problem, residual networks add identity mapping layers (Figure II.25) realized by adding shortcuts connections (skip connections) and element-wise addition to every few stacked layers [59].

ResNet was built by several stacked residual units and developed with many different numbers of layers: 18, 34, 50, 101, 152, and 1202. However, the number of the operations can be varied depending on the different architectures, the residual units are composed of convolutional, pooling, and layers. ResNet is similar to VGG net, but ResNet is about eight times deeper than VGG [60].



.

Figure II. 25. Shortcut connection.

II.5.2.5. VGG

In 2014, researchers at Oxford's Visual Geometry Group introduced two novel architectures named VGG16 and VGG19 [61]. VGG16 achieved a top-five accuracy rate of 91.90% in the ImageNet competition in 2014. The VGG16 architecture has 138,355,752 parameters, five convolution blocks, and three dense layers. Each block contains some convolutional layers and then a max pool layer to decrease the block output size and remove the noise. The main difference between VGG16 and VGG19 is that VGG19 has 19 convolution layers instead of

16 convolution layers. The number of parameters increases from 138,357,544 to 143,667,240 because of additional layers [62]. In Figure II.26, the general representation of VGGNet is illustrated. In Figures II.27 and II.28, the two variants of VGG16 and VGG19 are depicted.

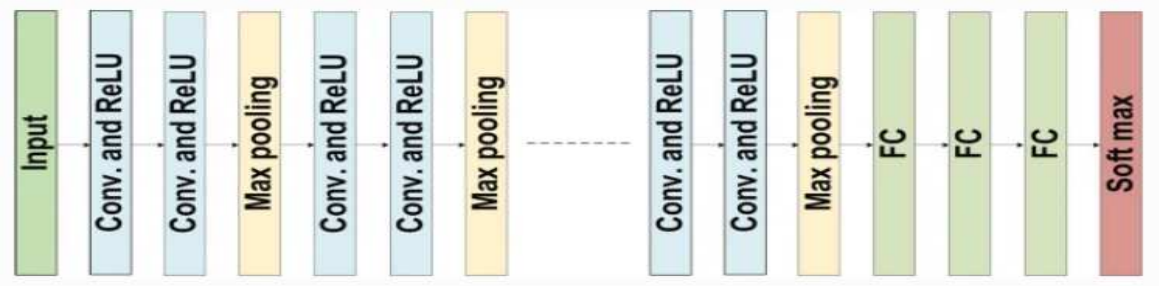


Figure II. 26. VGGNet Architecture.

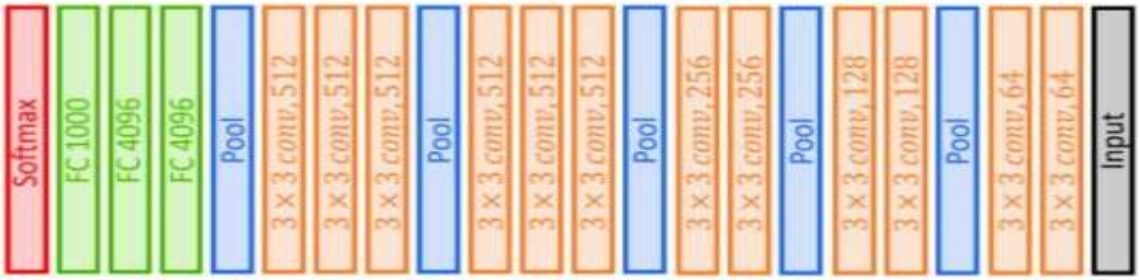


Figure II. 27. VGG 16 Model.

Figure II. 28. VGG 19 Model.

II.6. Importance and Controversy of DL & AI

Deep learning DL and artificial intelligence AI have gained significant attention and have become important areas of research and application in various fields. Here are some insights on the importance and controversies surrounding DL and AI:

a. Importance of DL & AI:

- Advanced automation: DL and AL enable the automation of previously manual operations, resulting in increased efficiency and productivity in industries such as manufacturing, logistics, healthcare, and finance.

- Data-driven decision making: DL and AL enable firms to analyze and process massive amounts of data in order to get useful insights and make data-driven decisions, resulting in better decision-making processes and outcomes.
- Increased personalization: DL and AI are being used in recommendation systems, tailored marketing, and content curation, allowing users, customers, and consumers to have more personalized experiences.
- Improved user experience: DL and AL are being utilized in user interfaces, virtual assistants, and chatbots to improve user experiences and customer service in a wide range of applications.
- Advancements in healthcare: DL and AI have the potential to transform healthcare by enabling early disease identification, individualized treatment regimens, and medical picture analysis, resulting in better patient care and results.

b. Controversies of DL & AI:

- Ethical issues: DL and AI are presenting ethical questions about prejudice, justice, and accountability because these technologies are built and trained using data that may reflect existing biases in society, thereby leading to discrimination and inequity.
- Privacy and security: DL and AI entail the collection and processing of massive amounts of data, raising issues about privacy, security, and data breaches, as well as the possible misuse of data for spying or other criminal purposes.
- Automation and job displacement: As DL and AI technologies enable greater automation, job displacement and unemployment in specific industries may occur, posing social and economic issues for employees and communities.
- Inadequate transparency: DL and AI models can be complex and difficult to understand.
- Potential for bias and discrimination: DL and AI models are vulnerable to biases in data and may unintentionally perpetuate discrimination, raising ethical and societal concerns about fairness, inclusion, and equity.

Deep learning and AI technologies have both positive and negative features, and it is critical to carefully analyze their ramifications and handle any disputes or ethical problems that may arise throughout their development and deployment. To maximize the benefits of deep learning and AI technologies while mitigating potential hazards, proper ethical considerations, transparency, and responsible use are required.

II.7. Conclusion

The study of and the use of artificial intelligence (AI) has grown significantly in recent years and has the potential to fundamentally alter many fields of endeavor and facets of society. Among many other advantages, AI technologies have increased automation, allowed data-driven decision making, improved personalization, improved user experience, and advanced healthcare. AI has the potential to revolutionize how we live and work by increasing the convenience, efficiency, and interconnectedness of our daily lives.

Nevertheless, there are a number of issues and debates in the field of AI that need to be addressed. These include privacy and security hazards, job displacement worries, lack of transparency, and the possibility of bias and discrimination. These issues must be resolved by ethically developing, deploying, and regulating AI technology while carefully weighing their moral ramifications.

In this chapter, we presented a brief overview of artificial intelligence and machine learning, as well as artificial neural networks. Then, we delved into the fundamentals of the field of deep learning, including convolutional neural networks. We provided an introduction to the main concepts of CNN, such as convolutions and pooling, and explained the different types of layers and their functions. Additionally, we listed several common transfer learning models. All of these concepts will enable us to proceed to the next chapter, where we will implement these theoretical notions. Specifically, we will use the basic CNN to develop a smart security system in real-time.

In the next chapter, we will focus on simulating various deep learning models. Our goal is to develop an embedded system equipped with a detection algorithm for an intelligent security system. This process will involve two main steps. In the first step, which will be discussed in detail in the next chapter, we will use Python programming on a GPU in Google Colab to train our model and determine the optimal parameters. Once the optimal model is selected, we will move on to the second step, which is the implementation phase. This step involves the online implementation of the model on the Nvidia Jetson Nano GPU card.

II.8. References cited in Chapter II

[1] D. Jakhar and I. Kaur, "Artificial intelligence, machine learning and deep learning: definitions and differences," *Clinical and experimental dermatology,* vol. 45, pp. 131-132, 2020.

[2] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, 2018, pp. 1-6.
[3] I. El Naqa and M. J. Murphy, *What is machine learning?*: Springer, 2015.
[4] T. M. Mitchell, *Machine learning* vol. 1: McGraw-hill New York, 2007.
[5] E. Alpaydin, *Introduction to machine learning*: MIT press, 2020.
[6] A. Garg and K. Rana, "WIRELESS COMMUNICATION AND MATHEMATICS," ed, 2020, pp. 183-193.
[7] T. M. K. S. Landset, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," *Journal of Big Data,* vol. 2, p. 24, 2015/11/05
[8] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet],* vol. 9, pp. 381-386, 2020.
[9] S. Doshi, "Commonsense Validation and Reasoning using Natural Language Processing," 2020.
[10] Ł. Chmielewski and L. Weissbart, "On reverse engineering neural network implementation on GPU," in *Applied Cryptography and Network Security Workshops: ACNS 2021 Satellite Workshops, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, and SiMLA, Kamakura, Japan, June 21–24, 2021, Proceedings*, 2021, pp. 96-113.
[11] M. Mishra and M. Srivastava, "A view of artificial neural network," in *2014 international conference on advances in engineering & technology research (ICAETR-2014)*, 2014, pp. 1-3.
[12] H. Kukreja, N. Bharath, C. Siddesh, and S. Kuldeep, "An introduction to artificial neural network," *Int J Adv Res Innov Ideas Educ,* vol. 1, pp. 27-30, 2016.
[13] Y.-S. Park and S. Lek, "Artificial neural networks: multilayer perceptron for ecological modeling," in *Developments in environmental modelling*. vol. 28, ed: Elsevier, 2016, pp. 123-140.
[14] A. A. Safar, D. M. Salih, and A. M. Murshid, "Pattern recognition using the multi-layer perceptron (MLP) for medical disease: A survey," *International Journal of Nonlinear Analysis and Applications,* vol. 14, pp. 1989-1998, 2023.
[15] G. A. Montazer and D. Giveki, "An improved radial basis function neural network for object image retrieval," *Neurocomputing,* vol. 168, pp. 221-233, 2015.
[16] H. Han, Q. Chen, and J. Qiao, "Research on an online self-organizing radial basis function neural network," *Neural computing and applications,* vol. 19, pp. 667-676, 2010.
[17] C. S. K. Dash, A. K. Behera, S. Dehuri, and S.-B. Cho, "Radial basis function neural networks: a topical state-of-the-art survey," *Open Computer Science,* vol. 6, pp. 33-63, 2016.
[18] S.-H. Yoo, S.-K. Oh, and W. Pedrycz, "Optimized face recognition algorithm using radial basis function neural networks and its practical applications," *Neural Networks,* vol. 69, pp. 111-125, 2015.
[19] I. G. Tsoulos and V. Charilogis, "Locating the Parameters of RBF Networks Using a Hybrid Particle Swarm Optimization Method," *Algorithms,* vol. 16, p. 71, 2023.
[20] V. Jakkula, "Tutorial on support vector machine (svm)," *School of EECS, Washington State University,* vol. 37, p. 3, 2006.
[21] A. Rashidi, "EVALUATING THE PERFORMANCE OF MACHINE-LEARNING TECHNIQUES FOR RECOGNIZING CONSTRUCTION MATERIALS IN DIGITAL IMAGES," master thesis,Georgia Institute of Technology, 2013.

[22] H. S. Ródenas, "Support Vector Machines for Survival Analysis:Methods and Variable Relevance," phd thesis,university of barcelona, 2017.
[23] L. Shen and A. Joshi, "An SVM-based voting algorithm with application to parse reranking," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003, pp. 9-16.
[24] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, pp. 121-167, 1998/06/01 1998.
[25] S. Agrawal, N. K. Verma, P. Tamrakar, and P. Sircar, "Content based color image classification using SVM," in *2011 Eighth International Conference on Information Technology: New Generations*, 2011, pp. 1090-1094.
[26] A. Mathur and G. M. Foody, "Multiclass and binary SVM classification: Implications for training and classification users," *IEEE Geoscience and remote sensing letters,* vol. 5, pp. 241-245, 2008.
[27] A. A. Elngar, M. Arafa, A. Fathy, B. Moustafa, O. Mahmoud, M. Shaban, and N. Fawzy, "Image classification based on CNN: a survey," *Journal of Cybersecurity and Information Management,* vol. 6, pp. 18-50, 2021.
[28] P. Kim, "Matlab deep learning with machine learning, neural networks and artificial intelligence," 2017.
[29] J. Wu, "Application of Artificial NeuralNetwork on Speech Signal Features for Parkinson's Disease Classification ", Master thesis,CALIFORNIA STATE UNIVERSITY SAN MARCOS, 2019.
[30] X. Du, Y. Cai, S. Wang, and L. Zhang, "Overview of deep learning," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 159-164.
[31] S. B. Srivallapanondh, "Dissecting the Performance of AI Applications Using NVIDIA GPUs at the Edge," Master's Thesis,aston university, 2021.
[32] S. Sumit. (2018). *A Comprehensive Guide to Convolutional Neural Networks - the eli5 Way*. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[33] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing,* vol. 13, p. 4712, 2021.
[34] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging,* vol. 9, pp. 611-629, 2018.
[35] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia computer science,* vol. 132, pp. 377-384, 2018.
[36] N. K. Chauhan and K. Singh, "A review on conventional machine learning vs deep learning," in *2018 International conference on computing, power and communication technologies (GUCON)*, 2018, pp. 347-352.
[37] P. Vallimeena, U. Gopalakrishnan, B. B. Nair, and S. N. Rao, "CNN algorithms for detection of human face attributes–a survey," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 576-581.
[38] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *ArXiv e-prints,* 2015.
[39] R. C. I. BENSAHA, "Radio Signal Classification using Deep Learning," master thesis,University of Ghardaia, 2021.

[40] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: history, architecture, application, challenges and future scope," *Electronics,* vol. 10, p. 2470, 2021.
[41] S. Hijazi, R. Kumar, and C. Rowen, "Using convolutional neural networks for image recognition," *Cadence Design Systems Inc.: San Jose, CA, USA,* vol. 9, p. 1, 2015.
[42] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data,* vol. 8, pp. 1-74, 2021.
[43] W. H. Lopez Pinaya, S. Vieira, R. Garcia-dias, and A. Mechelli, "Convolutional neural networks," in *Machine Learning*, ed: Elsevier, 2019, pp. 173-191.
[44] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2020, pp. 281-286.
[45] A. Sagar, "A New Activation Function for Training Deep Neural

Networks to Avoid Local Minimum," *Vellore Institute of Technology, Vellore, Tamil Nadu, India*.
[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.
[47] V. Balas, R. Kumar, and R. Srivastavs, *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, 2020.
[48] N. Ketkar and J. Moolayil, *Deep Learning with Python Learn Best Practices of Deep Learning Models with PyTorch*, second ed.: Apress, 2021.
[49] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378,* 2018.
[50] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in *2017 international conference on communication and signal processing (ICCSP)*, 2017, pp. 0588-0592.
[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, pp. 84-90, 2017.
[52] F. Özyurt, "A fused CNN model for WBC detection with MRMR feature selection and extreme learning machine," *Soft Computing,* vol. 24, pp. 8163-8172, 2020.
[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
[54] S. Verma, R. S. Tomar, B. K. Chaurasia, V. Singh, and J. Abawajy, *Communication, Networks and Computing: First International Conference, CNC 2018, Gwalior, India, March 22-24, 2018, Revised Selected Papers* vol. 839: Springer, 2018.
[55] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.
[56] A. Howard, A. Zhmoginov, L.-C. Chen, M. Sandler, and M. Zhu, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," 2018.
[57] M. Galanov, "Convolutional Neural Networks for 3D Action Recognition ", Master's Thesis, Masaryk University Faculty of Informatics, 2020.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[59] A. L. Vilardi, "VERY DEEP CONVOLUTIONAL NEURAL NETWORKS FOR FACE IDENTIFICATION," Master's Thesis, Universitat Politècnica de Catalunya, 2016.

[60] V. Maeda-Gutiérrez, C. E. Galvan-Tejada, L. A. Zanella-Calzada, J. M. Celaya-Padilla, J. I. Galván-Tejada, H. Gamboa-Rosales, H. Luna-Garcia, R. Magallanes-Quintanar, C. A. Guerrero Mendez, and C. A. Olvera-Olvera, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Applied Sciences,* vol. 10, p. 1245, 2020.

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[62] I. Hamdy and A. Kandel, "Deep Learning Techniques for Medical Image Classification," Master thesis,University Nova de Lisboa Lisbon, Portugal, 2021.

Chapter III

Simulation of Deep Learning Models

III.1. Introduction

Object detection is a research area within computer vision that has great practical value and many application prospects. These applications include smart video surveillance, robot navigation, automatic positioning in digital cameras, face detection and recognition, and human-computer interaction.

In the past, object detection was very limited due to a lack of data and low computing power. However, with the passage of time and the development of Graphic Processing Units (GPU) such as the NVIDIA Jetson family, high processing power is now available.

Object detection technology based on deep learning has developed rapidly, particularly YOLO (you only look once), which has become a state-of-the-art object detector for real-time applications. YOLOv5 is one of the latest versions of the YOLO algorithm that has gained popularity in the data science community due to its superior performance in challenging and noisy data environments, availability, and ease of use with widely-used programming languages such as Python.

As part of our work in building an intelligent security system, this chapter focuses on real-time weapons detection using the YOLOv5 model. However, before implementing the model on an embedded system, it is crucial to understand both the software and hardware prerequisites. In this chapter, we will specifically focus on the software aspect, namely the simulation of deep learning for real-time applications. This simulation will be used to inform the hardware implementation detailed in chapter four.

III.2. Software aspects

The most important aspects used in our implementation are presented in this section:

III.2.1. Python

Python is a modern, general-purpose, object-oriented, high-level programming language [1]. It is an open source programming language which means that we can freely download it and use it to develop programs. It can be downloaded from www.python.org.

Python is very flexible, because of its ability to use modular components that were designed in other programming languages [2].

It has libraries for data loading, visualization, statistics, natural language processing, image processing, and more. This vast toolbox provides data scientists with a large array of general and

special-purpose functionality. One of the main advantages of using Python is the ability to interact directly with the code, using a terminal or other tools like the Jupyter Notebook [3].

III.2.2. Tools and libraries

In this section, we present the tools and libraries we used in our system:

OS module: Python's OS module provides functions for interacting with the operating system. OS is a standard utility module in Python. This module provides a portable way of using operating system dependent functionality and many functions for interacting with the file system are included in the The *os* and *os.path* modules.

Jupyter Notebook: The Jupyter Notebook is a browser based development environment that enables bundling of code, text, and images. Jupyter Notebooks can be installed on Windows, Macintosh and Linux computers that have Python installed [4], it also makes it easy to share Notebooks with others and is optimal for collaborations [5].

NumPy: NumPy is one of the fundamental packages for scientific computing in Python. It provides multi-dimensional arrays and matrices, broadcasting functions, tools for integrating C/C++, Fortran code, mathematical, logical, shape manipulation, sorting, selecting, I/O, useful linear algebra, Fourier transform, random number capabilities, basic statistical operations and much more [1].

TensorFlow: TensorFlow is a powerful library for doing large-scale numerical computation. One of the tasks at which it excels is implementing and training deep neural networks. TensorFlow provides primitives for defining functions on tensors and automatically computing their derivatives [6].

The Google Brain team first designed the TensorFlow python deep-learning library for internal usage [7]. Lastly, allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API [6].

OpenCV: OpenCV is a cross-platform open source computer vision library, mostly employed for its real time image processing performance. It aims to provide well tested, optimized and open

source implementation of state of the art image processing and computer vision algorithms [8]. Opencv supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc.

Keras: Keras is a high-level API for neural networks in Python that can be used in combination with three different backends. These include TensorFlow, CNTK and Theano. With this framework, the focus of development is on a simple interface and on enabling rapid test execution. Keras supports both convolutional networks and recurrent networks and a combination of the two. The CPU or GPU can also be used for execution. The greatest advantages of Keras are the ease of use, the modularity and the simple expandability with new modules [9].

PyTorch: PyTorch is an open source machine learning library for deep learning, used for applications such as computer vision and natural language processing. This framework was developed by Facebook and it was created to provide models that are easier to write than other frameworks such as TensorFlow [10].

Matplotlib: Matplotlib is the primary visualization tool for scientific graphics in Python. Like all great open-source projects, it originated to satisfy a personal need [11], which produces scientific publication quality figures [12], such as line charts, histograms, scatter plots in a variety of hard copy formats and interactive environments across platforms.

Google Colab: Colaboratory is a Google research project, and it was created to help disseminate machine learning education and research [13].

Colaboratory notebooks are based on Jupyter and work as a Google Docs object: can be shared and users can collaborate on the same notebook. Also it's the infrastructure is hosted on the Google Cloud platform [14]. Google offers free use of GPU and it is an attractive feature to the developers. The reasons for making it publicly available could be to make its software a standard in the field of academia for teaching machine learning and data science. By using Colab, programmers could write, edit, and

execute code in python. Additionally, popular python libraries such as NumPy and Matplotlib could be used to analyze and visualize data. It also allows us to integrate open-source libraries named PyTorch, TensorFlow, Keras, and OpenCV [13].

When a user opens a Google Colab file, they must choose a runtime type. There are 3 available options for the same: None (that will use the computer's CPU the user is using), GPU, and TPU (especially for tensor processing). The selection box is found in Runtime -> Change runtime type and looks as shown in the image below.[15]

Figure III. 1. Google Colab Notebook Setting.

To start working with Colab we first need to log in to our google account, and then we go to this link https://colab.research.google.com. Then we press new notebook.

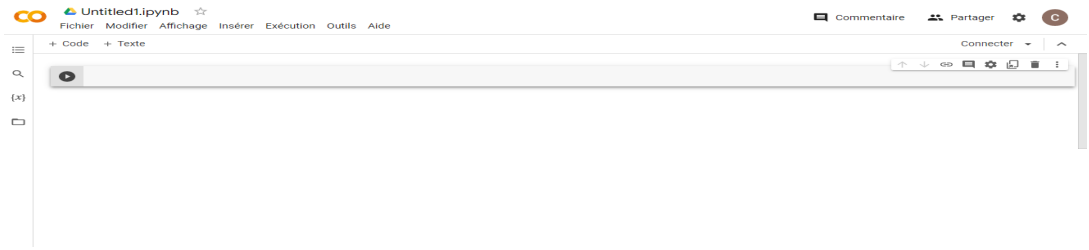


Figure III. 2. Google Colab File Name.

It will create a Jupiter notebook with Untitled0.ipynb and we save it to our Google drive in a folder named Colab Notebooks. Also we can create a new cell in Google Colab by pressing on + Code at the top of the notebook.

As we know commonly used python libraries are pre-installed and for news, we can install them using the following syntaxes:

!pip install (example: !pip install matplotlib-venn)

III.2.3. AI Accelerators

AI accelerators are specialized hardware designed to accelerate machine learning computations and improve performance in this section three popular options are presented.

III.2.3.1. Centralized Processing Unit (CPU)

CPU (central processing unit) is the primary component of the computer systems for performing arithmetic logic and control (I/O) specified instructions. the CPU consist of few cores that interacts with caches that can deal with few instructions at a time [16]. A massive collection of gadgets uses the CPU, with work locale, tablets, PC and workstations, Mobile phones even your smart TV. The most popular manufacturers are

AMD and INTEL work areas, PCs, and servers. At the same time, the enormous models of mobile phone used processor manufacturers are QUALCOMM, APPLE and NVIDIA [17].

III.2.3.2. Graphical Processing Unit (GPU)

GPU is a specialized computer hardware designed to accelerate parallel computing for image processing. Deep learning algorithms can benefit from GPU high parallelization to boost their performances, especially when dealing with visual data [18]. GPU's ability to offer peak performance makes them an ideal option for neural network acceleration. GPUs have small shallow caches that can be leveraged for parallel operations to compute a single function using vector processing units. This enables parallel operation's computation to execute a single piece of code in multiple pieces of data in parallel [19] NVIDIA offers a wide range of GPU products. The NVIDIA GPUs are very useful for machine learning engineers, as they provides a powerful programming platform and application programming interface (API) called Compute Unified Device Architecture (CUDA). CUDA works with a few of the most popular programming languages, such as Python, C, C++ [20].

III.2.3.3. Tensor Processing Unit (TPU)

Tensor Processing Unit is the recent development in ASICs by google, mainly designed for deep learning inference and training. These are the processors which are used in the data centers of Google Cloud to handle enormous amount of data .These specifically designed chips use a math library named as tensorflow framework, which is mainly used for deep learning and neural networks. Only some models of Google TPUs are available commercially because google TPUs are proprietary. These chips are designed to work on more input/output operations per joules with a large volume of low precision computing [21].

III.3. Object detection

Object detection is an important computer vision task that locates all positions of objects of interest in an input by bounding boxes and labeling them into categories that they belong to.

It can be treated as a combination of classification and localization, but multiple objects with different scales should be detected and classified at the same time within one image [22].

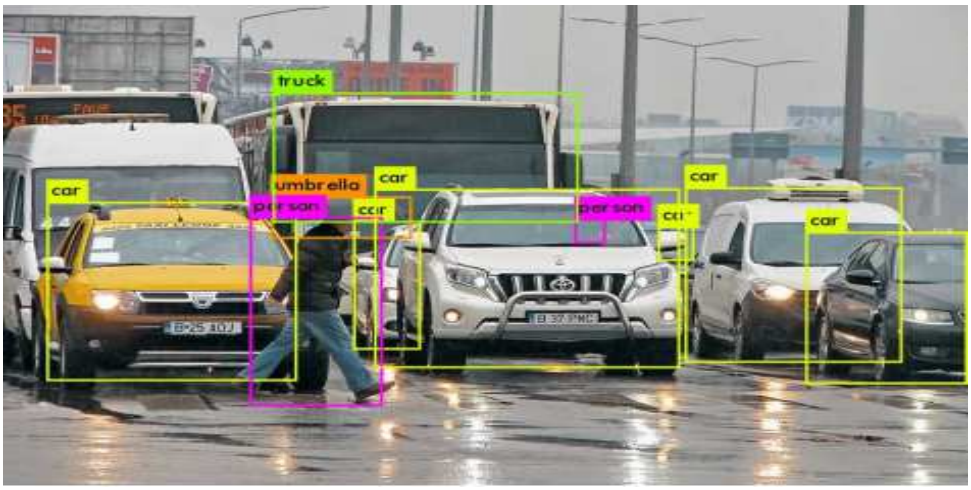


Figure III. 3. Object Detection.

Typical object detection models divide this task into three stages:

1. **Informative region selection:** Often performed using a sliding window approach, which performs evaluations and assigns a score over many rectangular subregions of an image. The location of the object is then defined as the subregion with the highest score.
2. **Feature extraction:** Identifies and extracts features such as color, texture, shape, or morphology that can make a representation of an object. Well extracted features increase the level of accuracy of the object detection model.
3. **Classification:** Predicts the class of each object present in the image [23].

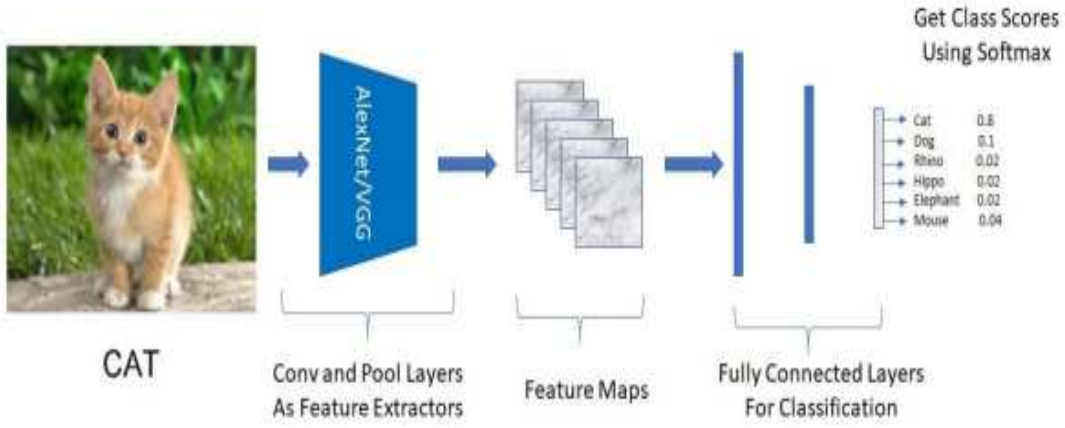


Figure III. 4. Classification with localization.

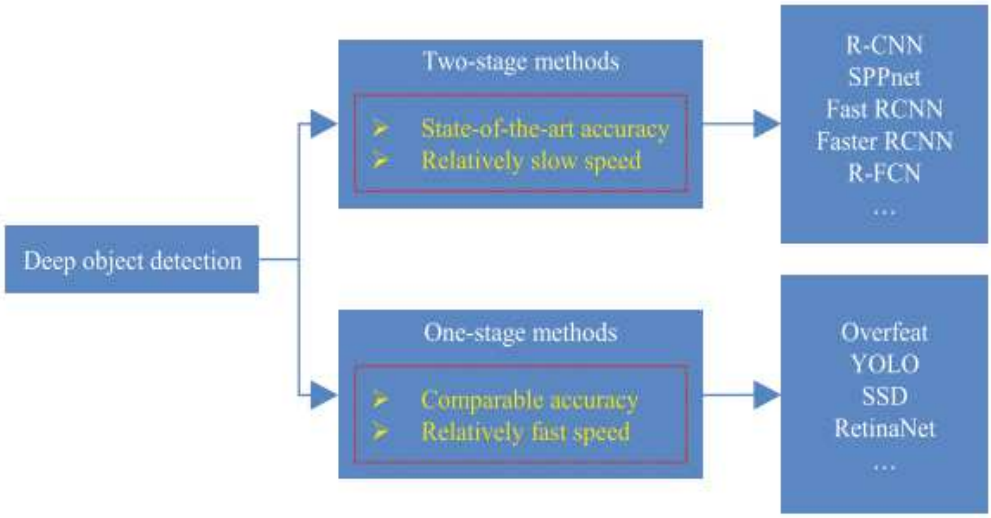


Figure III. 5. Categories of object detectors models.

As shown in Figure III.5, the approaches of object detection can be divided into two: approaches based on region proposal algorithms known as *Two-stage methods*. It uses a region proposal network to generate regions of interest in the first stage and sends region proposals to object classification and bounding box regression pipelines. The most representative ones, Faster R-CNN and FAST-RCNN [24, 25].

The second approach of object detection is *Single-stage methods*, where the detection process is considered "complete in one step", which mean single-stage object detectors make classification and place precise bounding boxes in one pass it recognized as real-time and unified networks such as YOLO (You Only Look Once) and SSD (Single Shot Detector) [24, 25].

In real-time processing, one-stage methods are faster compared to two-stage methods. For this reason, we are interested in using one-stage methods to build our system. According to scientific literature, YOLO is the most widely used method for real-time object detection. In the following subsection, we will discuss the YOLO method in detail.

III.3.1. YOLO

A You Only Look Once (YOLO) detector was proposed by Redmon et al [26] in 2016 and it is oriented to real-time processing. YOLO was inspired by GoogleNet and the idea was applying a unique neural network to the full image. Here the network divides the image into regions and simultaneously predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities [27].

Figure III. 6. A timeline of YOLO versions.

In December 2017, Joseph introduced another version of YOLOv2 with paper "YOLO9000: Better, Faster, Stronger." it was also known as YOLO 9000. After a year in April 2018, the most popular and stable version of YOLO was introduced. Joseph had a partner this time and they released YOLOv3 with paper "YOLOv3: An Incremental Improvement" [28].

In April 2020, YOLOv4 was introduced with some astounding new things, It outperformed YOLOv3 with a high margin and also has a significant amount of average precision when compared to EfficientDet Family [29].

The main idea of YOLO is to divide the input image into $K \times K$ cell grid. If the center of the object falls into one grid cell, the grid cell is set as responsible for predicting the object. YOLOv5 regards the detection task as a regression problem by using a single neural network to predict the bounding box and classes directly [30].

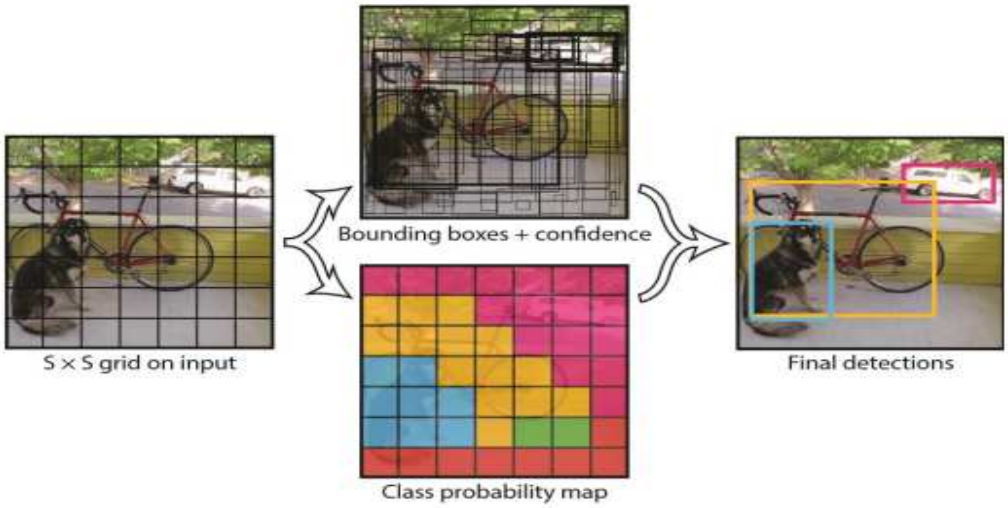


Figure III. 7. YOLO three step object detection process.

As shown in Figure III.8 each cell predicts B bounding boxes, which is composed of center, width, height and confidence, and C conditional class probabilities. YOLO predictions could be encoded as a $S \times S \times (B \times 5 + C)$ tensor for a given input image. However, all predicted bounding boxes do not belong to the objects in the image. Thus, predictions are needed to be filtered to get rid of duplicate and weak boxes. Predicted bounding boxes whose confidence score is lower than the predefined threshold could be ignored. After this elimination, there could still be some high confidence bounding boxes most probably belonging to the same object. To get rid of redundant bounding boxes, YOLO uses non-maximum suppression. Non-maximum suppression selects the bounding box with the highest confidence and suppresses all the remaining ones that have an overlapping ratio greater than the defined threshold.

Non-maximum suppression is done for each class, i.e., overlapping boxes that define different classes are not eliminated, and final detections are obtained [31].

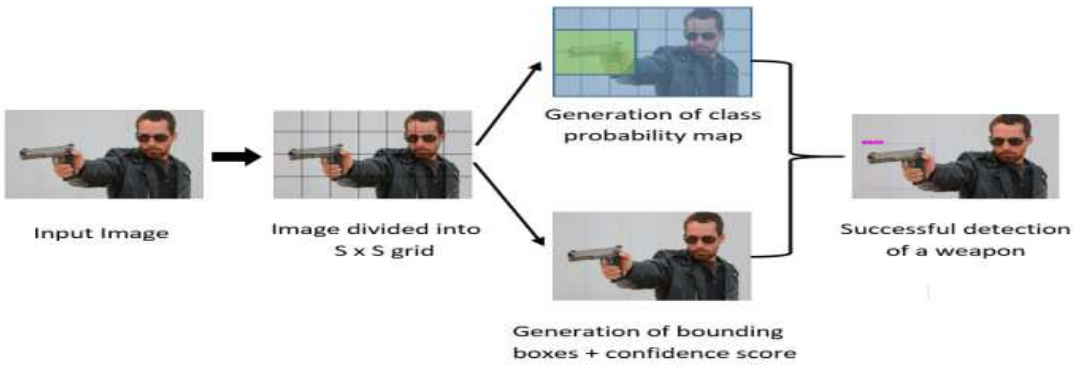


Figure III. 8. YOLO Architecture to Represent the Actions Performed on the Image.

There are multiple benefits associated with such architecture:

- **Detection speed:** YOLO is very fast - 100 times faster than even the modern R-CNN variants called Fast R-CNN. 1000 times faster than the traditional R-CNN approaches.
- **Global reasoning:** As YOLO models always process the image as a whole, they do not miss the opportunity to learn the contextual information about the environment in which the objects usually appear and use it in the detection process. Seeing the larger context, they are far less likely to mistake background patches for the objects.
- **Generalization representations:** The experiments proved that when trained on natural images and artwork, it outperforms other object detection approaches significantly.

Exhibiting these capabilities of generalization, one can assume YOLO models are far less likely to fail when facing extraordinary detection challenges [32].

III.3.2. YOLOv5 model

Glenn Jocher presented the one-stage target identification method known as YOLOv5 in 2020 [33]. It has been developed using the PyTorch library of Python [34] and offers a wide variety of different deployment options. It is based on a PyTorch implementation but also provides easy conversions to other formats like TensorFlow, ONNX (Open Neural Network Exchange), CoreML, TensorRT and OpenVINO . This makes it easy to deploy fine-tuned or pretrained YOLOv5 on almost any platform. The shallowest network structures can efficiently be used on mobile devices while the deeper options can be deployed on computation servers to provide simultaneous real-time object detection for multiple video streams [35].

Yolov5 balances detection accuracy and real-time performance, with detection speed of up to 140 frames per second [36].

YOLOv5 is the most advanced detection network of the YOLO object detection algorithm, Based on the YOLOv3 and YOLOv4 algorithms. The model size of the YOLOv5 network is approximately one-tenth that of the YOLOv4 network. It has faster recognition and positioning speeds, and the accuracy is no less than that of YOLOv4 [37].

YOLOv5 provides four different scales for their model, S, M, L and X which stand for Small, Medium, Large, and X large, respectively. Each of these scales applies a different multiplier to the depth and width of the model, meaning the overall structure of the model remains constant, but the size and complexity of each model are scaled [38].

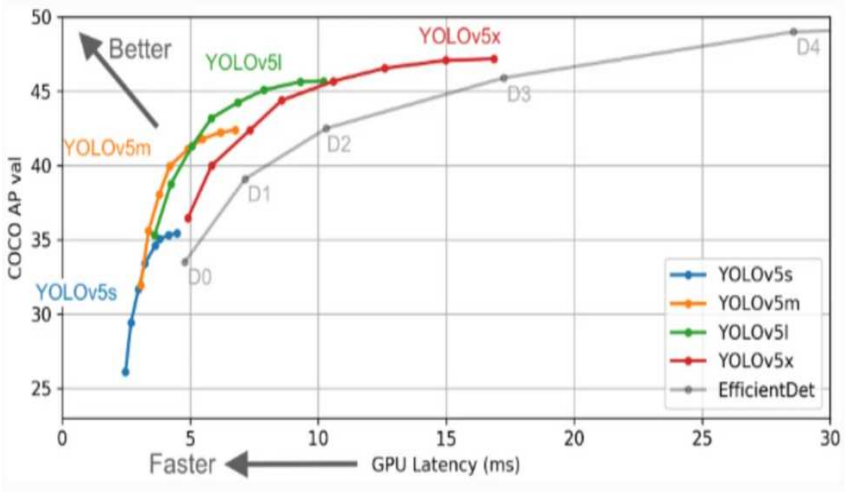


Figure III. 9. Versions of YOLOv5.

III.3.2.1. YOLOv5 Architecture

As shown in Figure III.10, the YOLOv5 model included three crucial components:

1) Backbone: the backbone is made of a CSPNet. The CSPNet reduces the model's complexity, resulting in fewer hyperparameters and FLOPS. At the same time, it resolves vanishing and exploding gradient issues, due to the depth of the neural networks. These improvements enhance inference speed and accuracy in object detection. Inside the CSPNet, there are several convolutional layers, four CSP bottlenecks with three convolutions, and spatial pyramid pooling. The CSPNet is responsible for extracting features from an input image and using convolutions and pooling to form a feature map that combines all extracted features. Thus, the backbone plays the role of feature extractor in YOLOv5.

2) Neck: the middle part of YOLOv5, often called the neck, is also known as the PANet. The PANet takes all the extracted features from the backbone and saves and sends them to the deep layers in order to perform feature fusions. These feature fusions are passed to the head so that high-level features are known to the output layer for final object detection.

3) Head: the head of YOLOv5 is responsible for object detection. It consists of 1x1 convolutions that predict the class of an object, with bounding boxes around the target object and a class probability score. Figure III.10 shows the overall architecture of YOLOv5 [39].

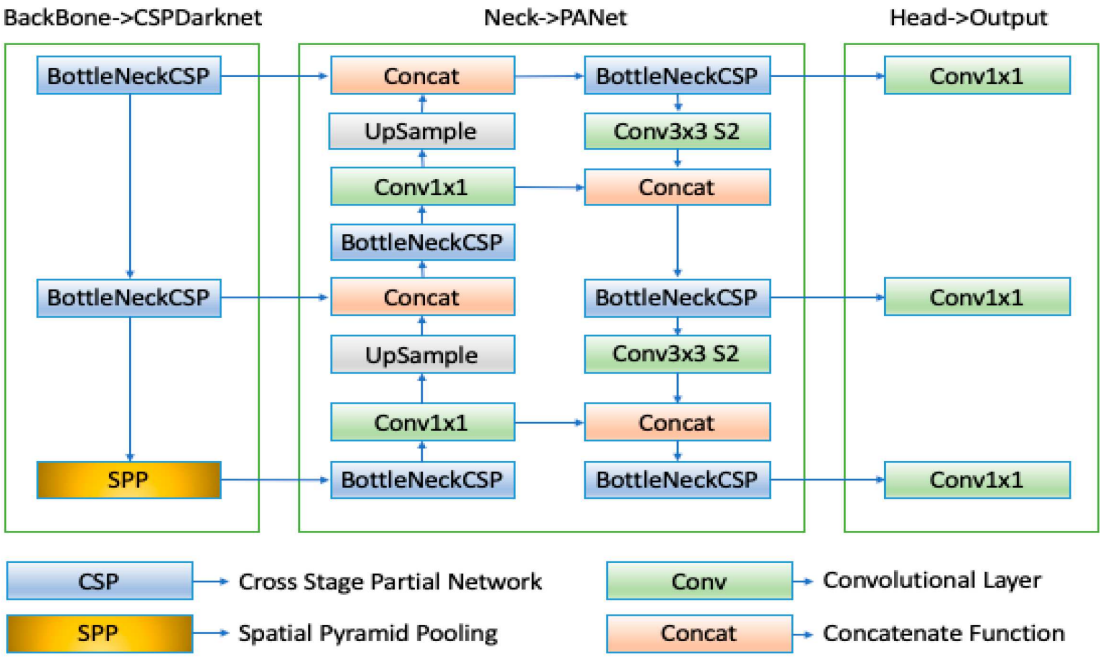


Figure III. 10. The YOLOv5 architecture.

III.3.2.2. Bounding-Box Regression and Loss Function

In object detection, Intersection over Union (IoU) is a standard for detecting object accuracy, which is used to measure the similarity between the predicted bounding box and the real bounding box, and can be described as Equation and Figure:

$$IoU = \frac{area(box(\Pr ed) \cap box(Truth))}{area(box(\Pr ed) \cup box(Truth))} \quad \text{(III.1)}$$

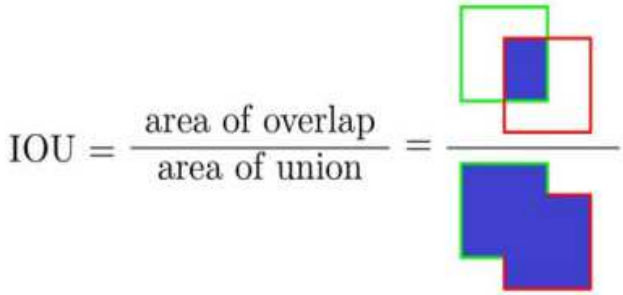


Figure III. 11. Illustration of intersection over union (IOU).

where the value range of IoU is [0, 1], which is a normalized index [40].

In Figure III.12 an example of a poor, good, and excellent IOU is presented:

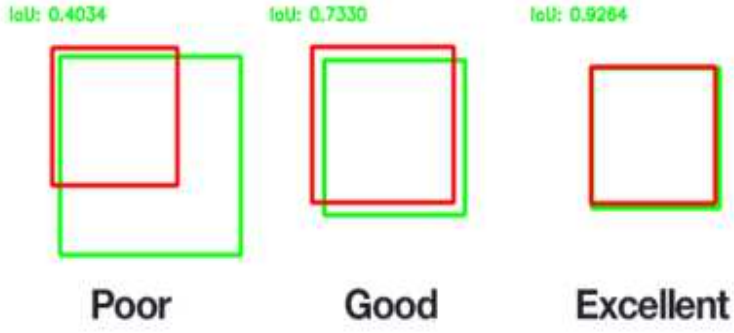


Figure III. 12. Illustration of IOU performance.

There is one problem with IoU. If there is no intersection between the two boxes, by definition, IoU equals zero. Therefore, it cannot reflect the distance (coincidence score) between the two. At the same time, since the loss is zero, there is no gradient returned, and learning and training cannot be performed. In paper [41], GIoU was come up, which is on the basis of IoU to solve the problem when the bounding boxes do not overlap. The equation is shown in [42].

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \setminus (A \cup B)|}{|C|} = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \quad \text{(III.2)}$$

where A and B are two bounding boxes of arbitrary shapes, C is the smallest rectangular box that can completely contain A and B, and the value range of GIoU is [−1, 1] [40].

Figure III. 13. GIoU evaluation diagram.

The loss function of YOLOv5 consists of classification loss, confidence loss, and regression loss of target prediction. The total loss can be expressed as follows:

$$L_{total} = \alpha l_{box} + \beta l_{obj} + \lambda l_{cls} \quad \text{(III.3)}$$

where l_{box} , l_{obj} and l_{cls} respectively represent target regression loss, target prediction confidence loss and category loss. Restricted by the optimizer, corresponding gain should be added to each type of loss in target detection to scale and balance each type of loss. In YOLOv5, the three gains α, β, and λ were determined to be 0.05, 1 and 0.5 by many trials [43].

III.3.3. Evaluation Metrics

Metrics are used to measure the performance of a neural network and allow for comparison between different models. Padilla et al [44].

Precision: This metric measures the percentage of relevant detection results. This can be determined using the following equation:

$$\text{Pr}\,ecision = \frac{TP}{TP + FP} \quad \text{(III.4)}$$

Where TP and FP denote the number of truly detected objects and the number of non-detected objects, respectively. Thus, this calculates the number of positive class predictions which belong to the positive class. This measures how much of the bounding-box predictions are true and correct [45].

Recall: The recall is the proportion of all targets that are correctly predicted, and it is governed by the following:

$$\text{Re}\,call = \frac{TP}{TP + FN} \quad \text{(III.5)}$$

where true positives (TPs) indicate the number of samples predicted by the algorithm as positive sample targets. False positives (FPs) indicate the number of samples that predict negative

samples to positive ones. False negatives (FNs) represent the number of samples that the algorithm predicts as positive samples relative to negative samples [46].

F1-Score: The F1-score is defined as the harmonic mean of the precision (Pr) and recall (Rc) of a given detector, The formula for the F1 score is shown below:

$$F1 = 2 \cdot \frac{\Pr ecision \cdot \mathrm{Re} call}{\Pr ecision + \mathrm{Re} call} \tag{III.6}$$

The F1-score is limited to the interval [0, 1], being 0 if precision or recall (or both) are 0, and 1 when both precision and recall are 1 [44].

Average Precision (AP): The AP represents the area enclosed by Precision and Recall curves. The formula is as follows: [47]

$$AP = \int_0^1 p(r)\,dr \tag{III.7}$$

Mean average precision (mAP): The mean Average Precision (mAP) is an evaluation metric used for object detection. Localization determines the position of an instance (the coordinates of the bounding box) and classification indicates what it is. The value of mAP is given by the following equation:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{III.8}$$

AP is the Average Precision, and N is the number of classes [48].

III.4. Methodology

After filtering out older models by reviewing related literature, papers, and conducting initial testing, we decided to work with the YOLO model. We found that it offers features that are highly relevant to our work, and we have conducted thorough testing with this model. Furthermore, working with YOLOv5 allows us to work with its sub-versions, including YOLOv5m, YOLOv5n, YOLOv5l, and YOLOv5s, and we have worked with most of them.

III.4.1. Data collection

Since crime rates increase in congested or suspicious-looking remote regions, security is typically the top issue in all companies. Right now, protecting person's existence from mass shootings is a concern. The most fundamental human right, the right to life, is at danger in response to mass shootings. The misery of mass shootings affects people's lives all across the world on a regular basis [49].

The use of weapons in public places has become a major problem in our society. These situations are more frequent in countries where weapons are legally purchased or their use is not controlled [50]. Every year, more than 15,000 people are killed in violent crimes, according to the World Health Organization. Knives and sharp cold steel weapons are used in around 40% of these homicides [51].

One way to reducing this kind of violence is prevention via early detection However, the detection of high-risk situations through Security cameras and video surveillance systems are still performed manually in many countries. It is a crucial and very tedious task if the longer duration is considered. Besides the limited performance of human may result in undetected dangers or delay in detecting threats, posing risks for citizen's security .An automatic weapon detection system based on artificial intelligence can provide the early detection of potentially violent situations that is of paramount importance for citizen's security. Deep Learning techniques based on Convolutional Neural Networks can be trained to detect presence of dangerous objects such as weapons in surveillance videos.

In our project, we present an automatic weapons detection system using deep learning mainly YOLOv5 algorithm which can effectively and accurately identify weapons.

III.4.2. Dataset

In machine learning and computer vision applications, datasets play a significant role because without good datasets the algorithms will not be able to perform well because it is the only way a model learns to see the world with an eye. Therefore, the collection of a good quality dataset is an important task [52].

The dataset that we used for training our model was published in Github website by Alberto Castillo Lamas and Fransco Pérez Hernandez [53, 54]. The datasets included in this section have been designed for the classification task based on CNN deep learning models.

The provided datasets are annotated by folders structure, where the images of a class are stored in a folder with the class name (two classes).

The datasets also attached the annotation files in Pascal VOC format with the region of the target objects in xml files.

1) Handgun detection: The Pistol detection dataset contains 2000 images of short guns with rich context in the background. The images selected from the internet contain one or more handguns in diverse situations including video surveillance contexts.

2) Knife detection: The Knife detection dataset contains 2000 images where at least one knife The dataset take into account:

a. Cold steel weapon of diverse types, shape, colors, size, and made of different materials.
b. Knives located at different distances near and far from the camera
c. Knives occluded partially by the hand.
d. Images captured in indoor and outdoor scenarios.

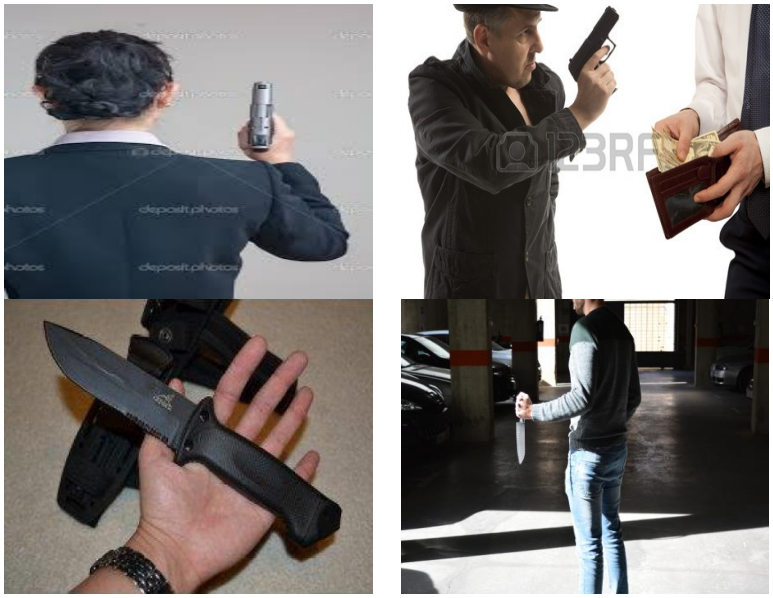


Figure III. 14. Dataset samples for handguns Class and knifes class.

We used 4000 images in the training set and 123 images for evaluating our model in the validation set. After preparing the dataset, we present the results of training and inference evaluation. We evaluated the custom models on our dataset using various evaluation metrics and recorded the results from the checkpoint that had the highest values. Next, we ran the models using different test images to evaluate the object detection performance and determine which model gave us the best accuracy and speed.

III.4.3. Implementing YOLOV5 Algorithm

YOLOv5 is a real-time object detector. Researchers have continuously improved the model by measuring its performance on the two official object detection datasets: Pascal VOC (visual object classes) and Microsoft COCO (common objects in context). YOLOv5 has four different models:

- YOLOv5n: This is the new nano model, smallest in the family and meant for the edge, IoT devices, and with OpenCV DNN support as well. It takes less than 2.5MB in INT8 format. It is ideal for mobile solutions.
- YOLOv5s: This is the small-sized model in the family. It has around 7.2 million parameters and is best suited for CPU inference.
- YOLOv5m: This is the medium-sized model in the family, it has 21.2 million parameters. It is an ideal model for many datasets and training since it provides a good tradeoff between speed and accuracy.
- YOLOv5l: It is the large model in the YOLOv5 family, it has 46.5 million parameters. It is ideal for datasets where target objects are small in size.
- YOLOv5x: It is the largest and has the highest mAP among the five models. It is slower compared to the others in terms of inference time and has 86.7 million parameters [55].

The Table III.1 shows a performance comparison made between these models on the Microsoft COCO dataset. In terms of speed, YOLOV5n is faster than other models.

Table III. 1. Details about the difference between Yolov5 versions.

Model	**Size (pixels)**	**mAP^{val} 0.5:0.95**	**mAP^{val} 0.5**	**Speed CPU b1 (ms)**	**Speed V100 b1 (ms)**	**Speed V100b 32 (ms)**	**Params (M)**	**FLOPs @640**
Yolov5n	640	28.4	46.0	45	6.3	0.6	1.9	4.5
Yolov5s	640	37.2	56.0	98	6.4	0.9	7.2	16.5
Yolov5m	640	45.2	63.9	224	8.2	1.7	21.2	49.0
Yolov5l	640	48.8	67.2	430	10.1	2.7	46.5	109.1
Yolov5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

III.4.3.1. Yolov5 Setup

The YOLOv5 algorithm is provided by Ultralytics and it is based on the PyTorch library. YOLOv5 coded in Python and hosted by the developers on the open-source platform Github.

We used Google Colaboratory to train the model since it provides free access to sophisticated GPUs without installation and also offers a seamless communication interface with the Google Drive where all the datasets and Yolov5 library stored in order to streamline the process as much as possible.

We import all the important libraries which will be used for the training of the model like (os, Tensorflow.keras, numpy…).

a) Mounting Our Personal Drive:

In order to use the dataset that we uploaded to the drive, we will mount our drive using the below code.

```
[ ]  from google.colab import drive
     drive.mount('/drive/')
```

A popup window will show up with the question "Permit this notebook to access your Google Drive files?" Select the blue Connect to Google Drive**.**

Figure III. 15. Connect to Google Colab Notebook to Google Drive.

Once the Drive is mounted, we'll get the message Mounted The root of our Google Drive will be mounted to /content/drive/My Drive/.

Then we have to download the dataset:

```
[ ]  #!git clone https://github.com/ari-dasci/OD-WeaponDetection.git
```

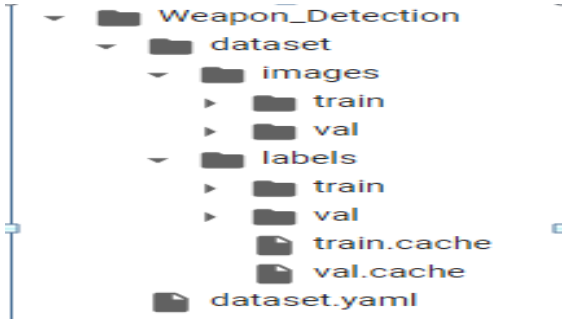


Figure III. 16. Folder structure for dataset.

The parent directory has three files and three sub-directories:

- data.yaml: It has the data-related configurations like the train and valid data directory path, the total number of classes in the dataset, and the name of each class.
- train: Training images along with training labels.

- valid: Validation images with annotations.
- test: Test images and labels. Assessing your model's performance becomes easy if test data with labels are available.

b) Installing the YOLOv5 Environment:

We need to clone the YOLOv5 repository. The next few lines of code clone the repository and enter into the yolov5 directory and install all the requirements that we may need for running the code.

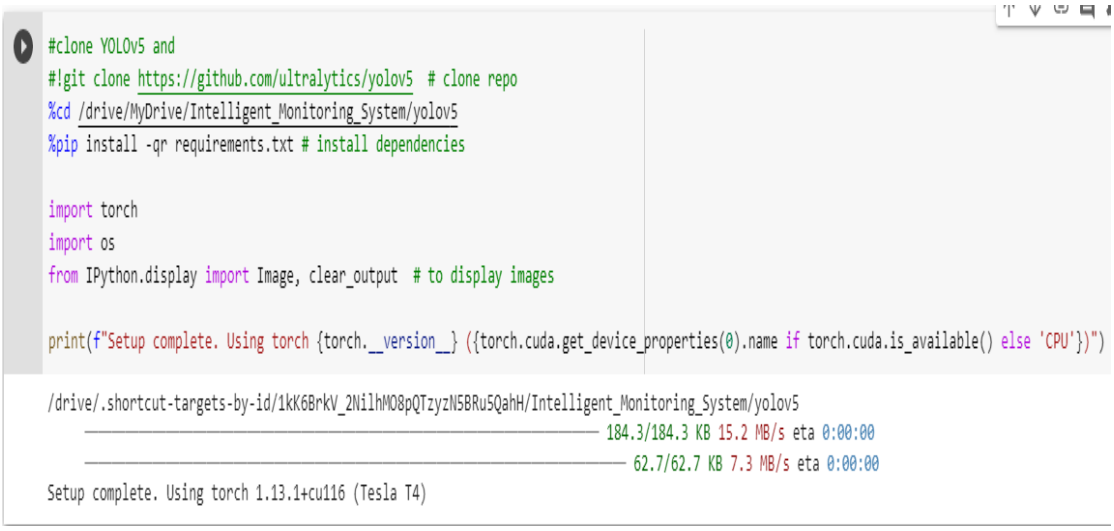


Figure III. 17. Cloning and installing the YOLOv5 repository.

c) Training YOLOv5 :

With the command line shown in Figure III.18, the model will be trained by compile file train.py along with its configurable arguments.

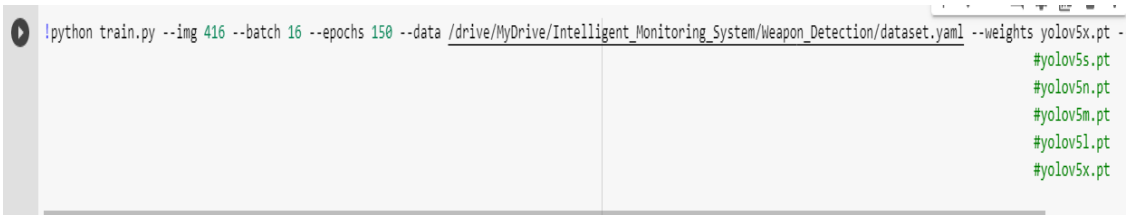


Figure III. 18. Implement the training process.

We define a few model parameters:

- **Img:** define input image size.
- **batch:** determine batch size.

- **Epochs:** define the number of training epochs.
- **Data:** set the path to our yaml file.
- **Cfg:** specify our model configuration.
- **Weights:** specify a custom path to weights.
- **Name:** result names.
- **Nosave:** only save the final checkpoint.
- **Cache:** cache images for faster training.

```
AutoAnchor: 4.50 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✓
Plotting labels to runs/train/exp2/labels.jpg...
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/exp2
Starting training for 150 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      0/149      6.71G    0.07054    0.02113    0.01672         48        416: 100% 250/250 [02:19<00:00,  1.79it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 4/4 [00:03<00:00,  1.30it/s]
                   all        123        153      0.488      0.565       0.49      0.258

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      1/149         8G    0.05174    0.01558   0.004852         39        416: 100% 250/250 [02:11<00:00,  1.90it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 4/4 [00:01<00:00,  2.32it/s]
                   all        123        153      0.438      0.588      0.476      0.253

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      2/149         8G    0.04755    0.01458   0.004169         36        416: 100% 250/250 [02:08<00:00,  1.94it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 4/4 [00:01<00:00,  2.43it/s]
                   all        123        153      0.454      0.605       0.52      0.278

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      3/149         8G    0.04511     0.0149    0.00448         35        416: 100% 250/250 [02:08<00:00,  1.94it/s]
```

Figure III. 19. Training progress in 100 epochs.

An add-in called TensorBoard was created that clearly visualized the entire training process.

```
[ ]  # Start tensorboard
     # Launch after you have started training
     # logs save in the folder "runs"
     %load_ext tensorboard
     %tensorboard --logdir runs
```

Figure III. 20. Use TensorBoard to load the entire training process saved in runs folder.

Besides, the model saves 2 weighting results as pt file.last.pt is the weight at the last epoch and best.pt file is the weight at the last epoch for the highest accuracy.

d) Inference with trained weight:

The inference session of the notebook that's where we can test the model we've trained the method of performing object detection with trained weights is similar to training the model.

Using the command shown in Figure II.21, detect.py file will be compiled, and it rebuilds the architecture used in the training. Trained weights will be used to predict objects and limit boxes.

```
[ ] %cd /drive/MyDrive/Intelligent_Monitoring_System/yolov5/
    !python detect.py --weights /drive/MyDrive/Intelligent_Monitoring_System/yolov5/runs/train/exp_yolov5x/weights/best.pt --img 416 --conf 0.1 --source /drive/

[ ] from google.colab import drive
    drive.mount('/content/drive')

[ ] #display inference on ALL test images

    import glob
    from IPython.display import Image, display

    for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
        display(Image(filename=imageName))
        print("\n")
```

Figure III. 21. Implement the detection process.

After the detection is complete, the predicted bounding boxes that cover the objects (weapons) will be drawn into the image. They will be saved in the same folder containing results from the training phase. Path would be like (runs/detect/exp).

Besides, the model saves 2 weighting results as pt file.last.pt file is the weight at the last epoch and best.pt file is the weight at the last epoch for the highest accuracy.

III.5. Results and discussions

In the same way we trained 100 epochs on the YOLOv5s, YOLOv5m, YOLOv5l, and the YOLOv5n and YOLOv5x.

Figures III.22 and III.23 and Table III.2 depict the results of our custom model training and validation on the weapons dataset, where we evaluated the accuracy of the models by measuring precision and recall at each iteration. We observed an increase in the evaluation metrics throughout the training epochs, suggesting that the models were learning and improving. The metrics remained relatively stable over the training epochs, with only minor fluctuations indicating that the models were converging.

The training and validation sets were evaluated for each iteration, with three different types of loss shown in Figure III.22: Box loss, Objectness loss, and Classification loss. The box loss

measures how well the algorithm can locate the center of an object and how well the predicted bounding box covers the object. Objectness is a measure of the probability that an object exists in a proposed region of interest. If the objectness is high, it means that the image window is likely to contain an object. The classification loss indicates how well the algorithm can predict the correct class of a given object.

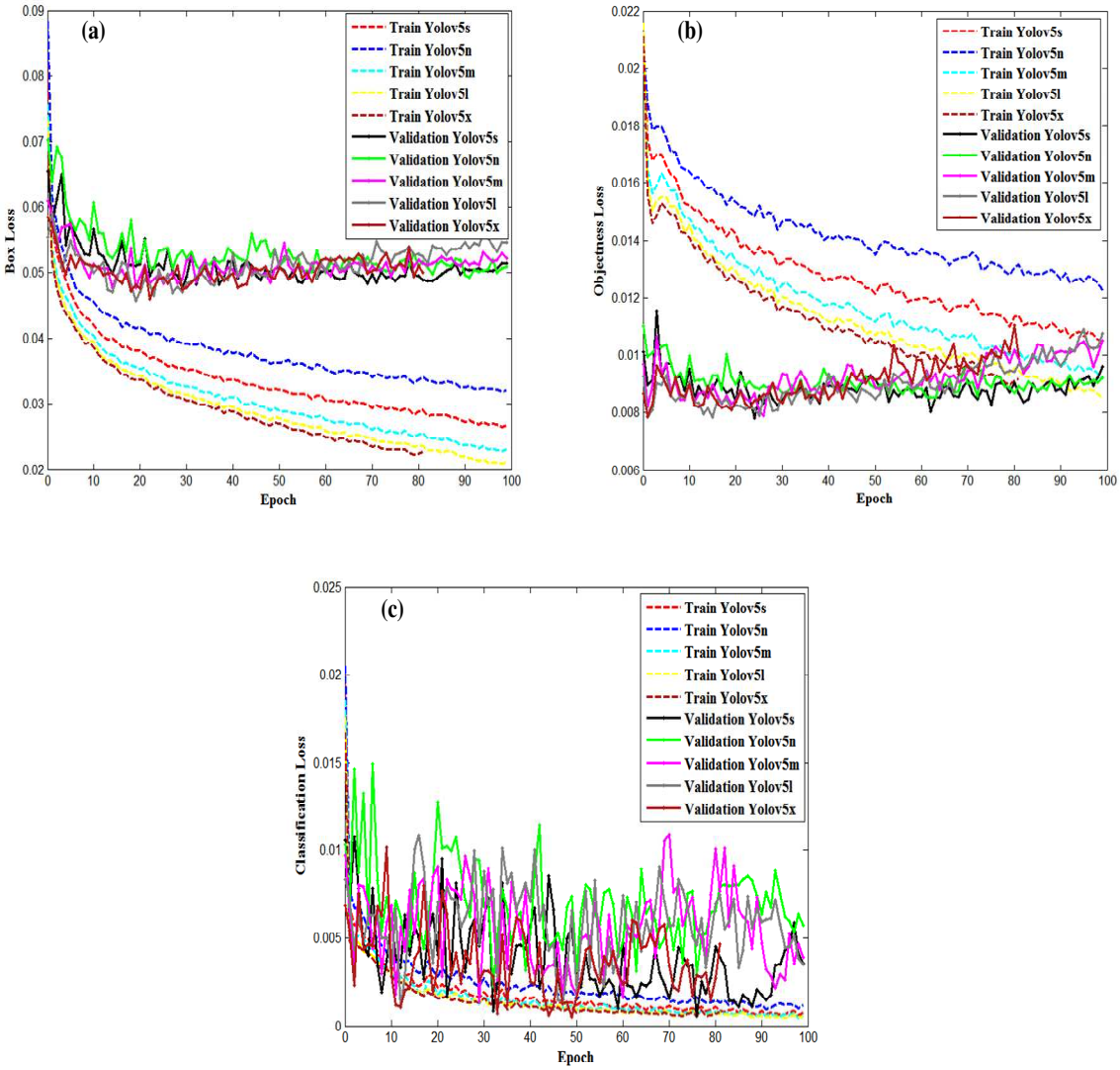


Figure III. 22. Results of evaluation Metrics: (a) box loss, (b) the objectness loss, and (c) the classification loss.

Figure III.22 represents the graphic results of the exploration consisting of box loss, classification loss, and object loss. The three graphics are the result of loss after doing 100 epochs for training and validation. In all three, it is shown that the loss value tends to continue to decrease. These results mean that the three classes: knife and handgun, which we use for detection, are accurately recognized during the training process.

- The box loss decrease steadily and almost stabilizing around the 40th epoch. The validation box loss values are better than the training values and continue to improve. It could indicate that the models predict well the location of objects.
- While the objectness loss decrease slowly over time The validation objectness loss has a slightly worse curve than the training one which indicates that the models are having a harder time predicting the presence of weapons (knifes –handguns) rather than locating the weapons knifes –handguns) themselves.
- The training classification loss drops rapidly as well as for the validation loss indicating that the learning efficiency of the models is high the models can predict the correct class of a given object.

To evaluate the performance of our model, we calculated the precision and recall for each iteration on the validation set, which are presented in Figures III.23-(a) and III.23-(b). The results indicate that the YOLOv5l model outperforms the other models in terms of precision and recall. Additionally, the YOLOv5l model generated higher values for the mean average precision at a threshold of 0.5 (mAP 0.5) and mean average precision at a threshold of 0.5 to 0.95 (mAP 0.5:0.95), as illustrated in Figures III.23-(c) and III.23-(d).

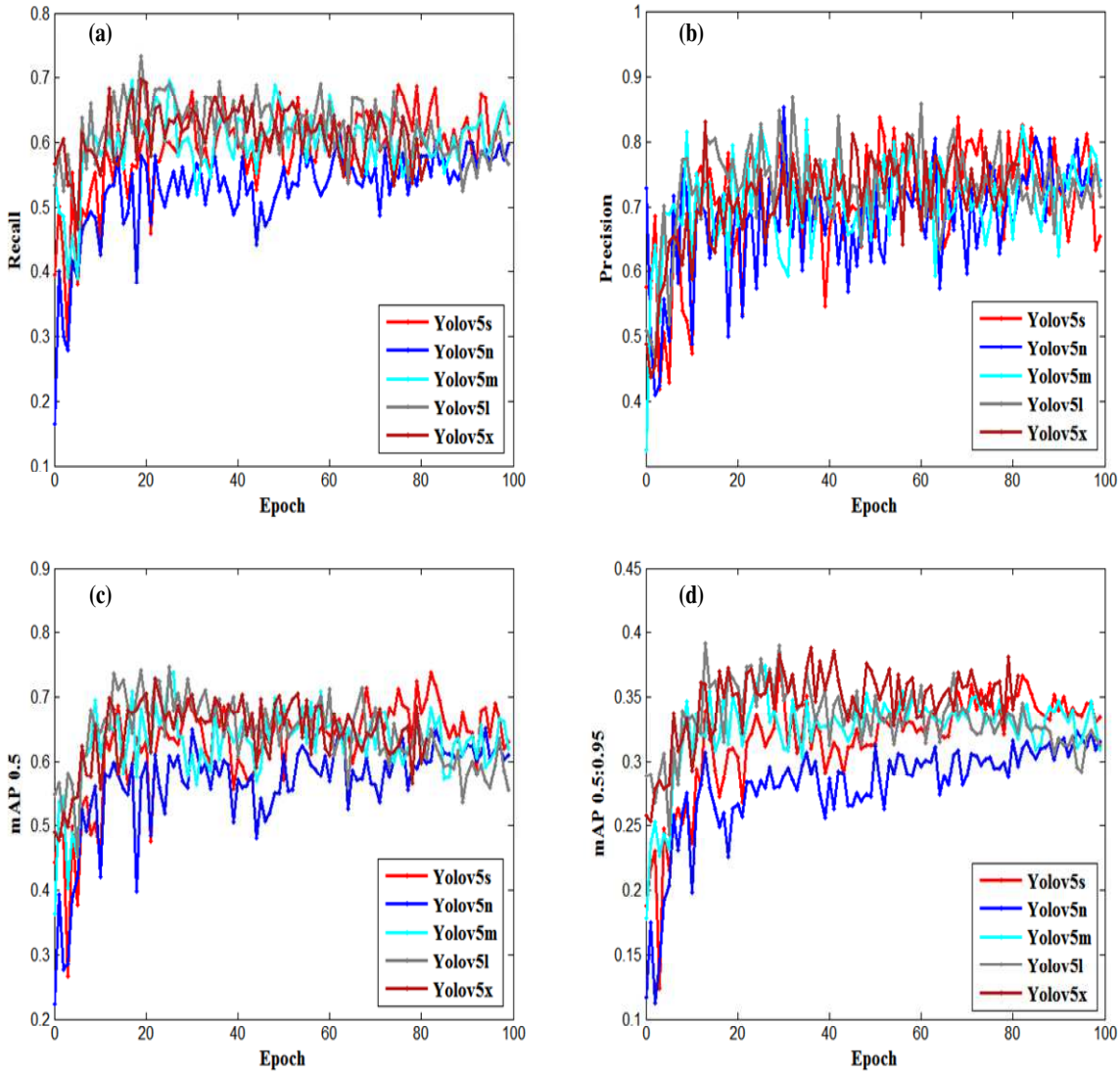


Figure III. 23. Results of evaluation Metrics: (a) Recall, (b) Precision and Mean Average Precision. (c) mAP 0.5, and (d) mAP 0.5:0.95.

After completing the training of the models, we evaluated the versions with the best results on the validation dataset. As shown in Table III.2, the YOLOv5 model provided better weapons detection accuracy, with higher precision, recall, mAP 0.5, and mAP 0.5:0.95 values of 86.8%, 73.2%, 74.5%, and 39.1%, respectively. The Objectness Loss Validation was lower for YOLOv5s. Additionally, the YOLOv5l model performed better in terms of Loss Training Box (2.09%), Loss Training (4.5%), and Objectness Loss Training (0.8%) than the other models. The

YOLOv5x model provided the lowest value for Classification Loss Validation at 0.05%, while YOLOv5m gave the best value for Classification Loss Training at 0.04%.

Table III. 2. Results of different evaluation criteria using YOLO models on the training and validation dataset.

Methos	Yolov5s	Yolov5n	Yolov5m	Yolov5l	Yolov5x
Box Loss Training	0.02648	0.03196	0.02281	**0.02092**	0.02226
Box Loss Validation	0.04746	0.04919	0.04771	**0.04575**	0.04595
Objectness Loss Training	0.01040	0.01225	0.00917	**0.00849**	0.00897
Objectness Loss Validation	**0.00780**	0.00853	0.00787	0.00782	0.00785
Classification Loss Training	0.00061	0.00094	**0.00047**	0.00048	0.00057
Classification Loss Validation	0.00055	0.00229	0.00143	0.00104	**0.00052**
Precision	0.83778	0.85341	0.83477	**0.86862**	0.83066
Recall	0.68915	0.61288	0.69594	**0.73289**	0.69743
mAP 0.5	0.73737	0.64980	0.73713	**0.74556**	0.72638
mAP 0.5:0.95	0.37269	0.32246	0.37350	**0.39155**	0.38823

(Note: The lower loss the better model; the higher matrices the better model)

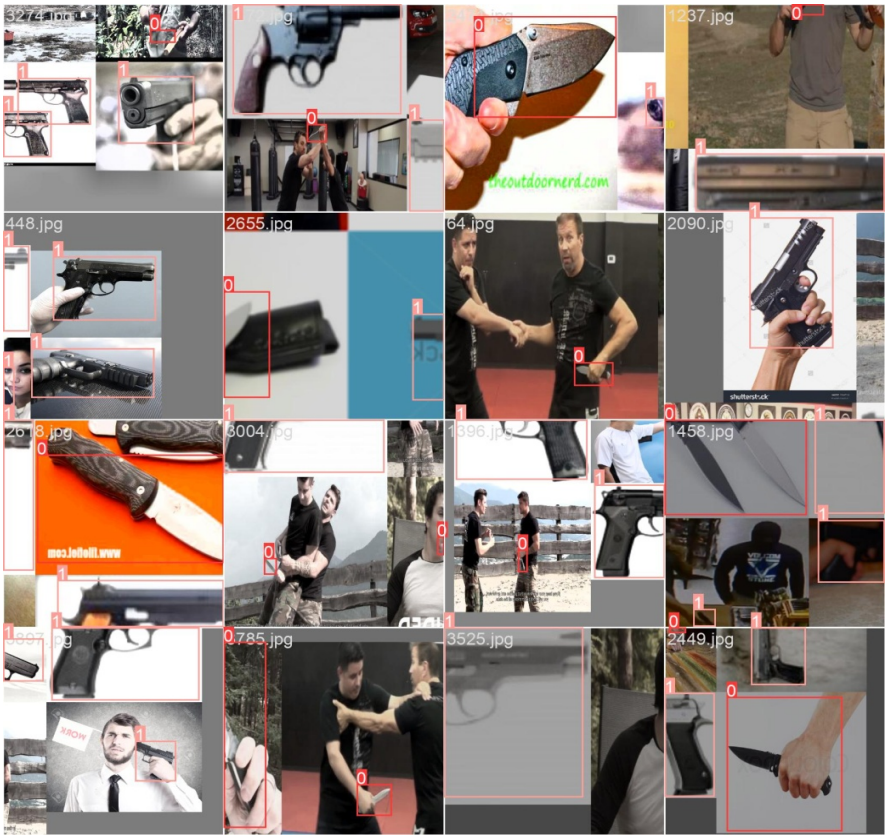


Figure III. 24. Detection results of weapons classes; handguns and knives for YOLOv5n.

As shown in Figure III.24, our custom YOLOv5n model demonstrated its ability to successfully detect and classify weapons in various test images from the weapons dataset, precisely recognizing the two classes of 'handguns' and 'knives'. Even in challenging cases where the images were affected by glare or shadows, the model was able to accurately identify the weapons. It should be noted that the images presented in the figure are just a sample of the situations encountered during the testing phase, and the model's performance may vary in other scenarios.

Recommended Model: The presented models, including YOLOv5s, YOLOv5l, YOLOv5m, YOLOv5n, and YOLOv5x, demonstrated good performance in terms of accuracy for weapons detection. However, since real-time processing and quick response times are critical for a weapons detection system, we chose to use the YOLOv5n model as the basis for our project. We believe that this model strikes the best balance between speed and accuracy, making it ideal for our needs.

III.6. Conception of an intelligent surveillance system

The surveillance area has always been an active area. Since ancient times, security measures have always been important for the protection and survival of the mankind. During the passage of time, man has always been inventing ways to detect risk situations more effectively, using equipment such as: audible systems, like bells; visual elements such as movement ropes, torches, or signals emitted by people; and lately vision equipment, which allows it to detect strange people [56].

Intelligent surveillance system (ISS) is a surveillance system that has intelligent capability to automatically analyze surveillance data and perform necessary actions such as generating alarm or warning. ISS is interdisciplinary topic that involves electronic (sensing device), computer vision and pattern recognition, artificial intelligence (machine learning), networking, communication and other areas. Intelligent surveillance system is promising to be implemented in various environments and applications. Some typical applications are listed as follows [57].

- Transport applications such as airports, maritime environments, railways, underground, and motorways to survey traffic.
- Public places such as banks, supermarkets, homes, department stores and parking lots.
- Remote surveillance of human activities such as attendance at football matches or other activities.

- Surveillance to obtain certain quality control in many industrial processes, surveillance in forensic applications and remote surveillance in military applications [58].

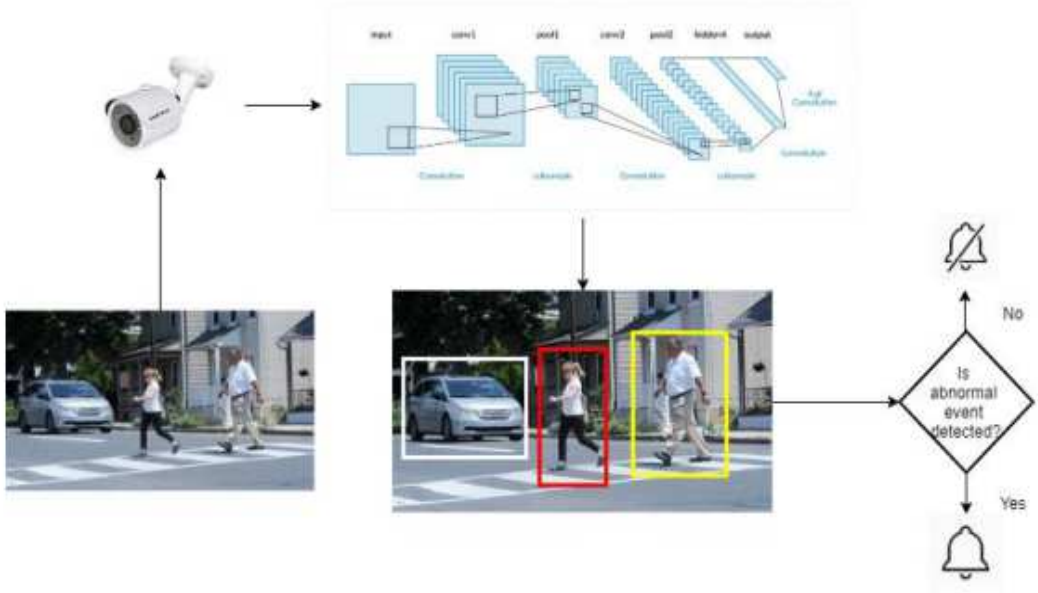


Figure III. 25. Normal flow of intelligence video surveillance systems.

As shown in Figure the input video frame from the video surveillance system is fed into the convolutional neural network object detector. The features of the input image such as texture, edge, shape, motion of are extracted for the later layers to identify the existence of objects in the image. Based on the existence of objects or the activities, the system will raise alarm to inform the detection of abnormal events [59].

Surveillance systems have been implemented throughout the years on various devices and infrastructures, mostly being CPU and FPGA based embedded systems. The more powerful and much efficient GPU has been used to implement the principles of surveillance mechanism with the traditional algorithms of face recognition and object detection and for the past few years, several deep learning architecture has shown a promising accuracy on the GPU based system [60]. In our project, a smart monitoring system that employs deep learning techniques. We will implement it on a powerful and efficient GPU Nvidia Jetson Nano.

III.7. Conclusion

In recent years, with the advancement of Deep Learning in object detection, multiple deep detection models are proposed. Throughout this chapter we have discussed one of the most

important visual tasks which are the object detection with its two main classes: the two-stage and one-stage methods, then we have focused on the YOLOv5 architecture.

In This chapter we aimed to determine the best models that can be used later for real-time application which is weapons detection, we discussed and compared the results obtained after training of each version of YOLOv5 model. We determined which model would give us the highest accuracy and also which model would be perfect considering a real-time implementation. Based on the information detailed in this chapter, we found that YOLOv5n is the suitable model.

Overall, in this chapter, we have discussed the first phase of building an intelligent surveillance system, which involves parameterizing the model and simulating the different steps in offline processing. This step requires significant hardware power, which is why we utilized Google Colab with a GPU to simulate our problem. In the next chapter, we will focus on the hardware aspect of our project. We will implement the best model found in the first step that combines accuracy and speed, and deploy it on an embedded system for real-time applications. For this purpose, we will use the powerful Nvidia Jetson Nano GPU card.

III.8. References cited in Chapter III

[1] R. Kumar, "Future for scientific computing using Python," *International Journal of Engineering Technologies and Management Research,* vol. 2, pp. 30-41, 2015.

[2] M. Nosrati, "Python: An appropriate language for real world programming," *World Applied Programming,* vol. 1, pp. 110-117, 2011.

[3] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python A Guide for Data Scientists*. United States of America: O'Reilly Media, Inc.. 2016.

[4] P. A. Craig, J. A. Nash, and T. D. Crawford, "Python scripting for biochemistry and molecular biology in Jupyter Notebooks," *Biochemistry and Molecular Biology Education,* vol. 50, pp. 479-482, 2022.

[5] D. Rolon-Mérette, M. Ross, T. Rolon-Mérette, and K. Church, "Introduction to Anaconda and Python: Installation and setup," *Quant. Methods Psychol,* vol. 16, pp. S3-S11, 2016.

[6] J. Lawrence, J. Malmsten, A. Rybka, D. A. Sabol, and K. Triplin, "Comparing TensorFlow deep learning performance using CPUs, GPUs, local PCs and cloud," 2017.

[7] A. K. Suzon, "FACE MASK DETECTION IN REAL TIME USING PYTHON," Centria University of Applied Sciences, 2022.

[8] B. Thorne and R. Grasset, "Python for prototyping computer vision applications," 2010.

[9] S. Jamal Agha, "Object Detection using YOLOv3," master thesis,, 2021.

[10] A. Malta, M. Mendes, and T. Farinha, "Augmented reality maintenance assistant using yolov5," *Applied Sciences,* vol. 11, p. 4758, 2021.

[11] J. Unpingco, "Getting Started with Scientific Python," in *Python for Probability, Statistics, and Machine Learning*, ed: Springer, 2022, pp. 1-46.

[12] N. Ari and M. Ustazhanov, "Matplotlib in python," in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 2014, pp. 1-6.

[13] S. Ray, K. Alshouiliy, and D. P. Agrawal, "Dimensionality reduction for human activity recognition using google colab," *Information,* vol. 12, p. 6, 2020.

[14] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. Reboucas Filho, "Performance analysis of google colaboratory as a tool for accelerating deep learning applications," *IEEE access,* vol. 6, pp. 61677-61685, 2018.

[15] P. Kanani and M. Padole, "Deep learning to detect skin cancer using google colab," *International Journal of Engineering and Advanced Technology Regular Issue,* vol. 8, pp. 2176-2183, 2019.

[16] Z. Memon, F. Samad, Z. Awan, A. Aziz, and S. Siddiqi, "CPU-GPU Processing," *International Journal of Computer Science and Network Security,* vol. 17, pp. 188-193, 2017.

[17] P. Raj and C. Sekhar, "Comparative Study on CPU, GPU and TPU," *International Journal of Computer Science and Information Technology for Education,* vol. 5, pp. 31-38, 2020.

[18] Ł. Chmielewski and L. Weissbart, "On reverse engineering neural network implementation on GPU," in *Applied Cryptography and Network Security Workshops: ACNS 2021 Satellite Workshops, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, and SiMLA, Kamakura, Japan, June 21–24, 2021, Proceedings*, 2021, pp. 96-113.

[19] L. Kalyanam, "Edge Computing for Deep Learning-Based Distributed Real-time Object Detection on IoT Constrained Platforms at Low Frame Rate," University of South Florida 2020

[20] Y. Deng, "Deep learning on mobile devices: a review," in *Mobile Multimedia/Image Processing, Security, and Applications 2019*, 2019, pp. 52-66.

[21] A. Shahid and M. Mushtaq, "A survey comparing specialized hardware and evolution in TPUs for neural networks," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1-6.

[22] H. Zhang and X. Hong, "Recent progresses on object detection: a brief review," *Multimedia Tools and Applications,* vol. 78, pp. 27809-27847, 2019.

[23] D. L. Kristofer Klarin, "Object and Anomaly Detection," Bachelor Thesis,Örebro University, 2022.

[24] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, "An evaluation of deep learning methods for small object detection," *Journal of electrical and computer engineering,* vol. 2020, pp. 1-18, 2020.

[25] G. A. Benslimane Echyma "Vision based vehicle counting at crowded intersections," master thesis,UNIVERSITY OF MOHAMED BOUDIAF - MSILA, 2022.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[27] Á. Morera, Á. Sánchez, A. B. Moreno, Á. D. Sappa, and J. F. Vélez, "SSD vs. YOLO for detection of outdoor urban advertising panels under multiple variabilities," *Sensors,* vol. 20, p. 4587, 2020.

[28] M. Rajput. (2020). *YOLOv5 is Here! Elephant Detector Training Using Custom Dataset & YOLOV5*. Available: https://towardsdatascience.com/yolo-v5-is-here-b668ce2a4908

[29] I. MAHDJOUBI, "Object Detection For Quadrotor Using Deep Learning," Master thesis UNIVERSITY OF M'SILA, 2022.

[30] F. Xie, B. Lin, and Y. Liu, "Research on the coordinate attention mechanism fuse in a YOLOv5 deep learning detector for the SAR ship detection task," *Sensors,* vol. 22, p. 3370, 2022.
[31] T. AKSOY, "IMPROVEMENTS ON ONE-STAGE OBJECT DETECTION BY VISUAL REASONING," Master thesis,Middle East Technical University, 2022.
[32] M. Láníček Adam, "Classification of Radar Detections Using Convolutional Neural Networks," Bachelor's Thesis,Brno University of Technology, 2020.
[33] M. Khalid, M. S. Sarfraz, U. Iqbal, M. U. Aftab, G. Niedbała, and H. T. Rauf, "Real-Time Plant Health Detection Using Deep Convolutional Neural Networks," *Agriculture,* vol. 13, p. 510, 2023.
[34] Y. A. Bomantara, H. Mustafa, H. Bartholomeus, and L. Kooistra, "Detection of Artificial Seed-like Objects from UAV Imagery," *Remote Sensing,* vol. 15, p. 1637, 2023.
[35] S. Pohjola, "OBJECT DETECTOR FINE-TUNING FORCOMPUTER VISION APPLICATIONS
In surveillance and adverse condition data domains," Master Thesis,Tampere University, 2022.
[36] H. M. Suliman and S. Kivrak, "Anti-Tank Guided Missile System Design Based on an Object Detection Model and a Camera," *International Journal of Computational Intelligence Systems,* vol. 16, p. 20, 2023.
[37] Z. Chen, R. Wu, Y. Lin, C. Li, S. Chen, Z. Yuan, S. Chen, and X. Zou, "Plant disease recognition model based on improved YOLOv5," *Agronomy,* vol. 12, p. 365, 2022.
[38] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," *arXiv preprint arXiv:2112.11798,* 2021.
[39] B. Aydin and S. Singha, "Drone Detection Using YOLOv5," *Eng,* vol. 4, pp. 416-433, 2023.
[40] Q. Song, S. Li, Q. Bai, J. Yang, X. Zhang, Z. Li, and Z. Duan, "Object detection method for grasping robot based on improved YOLOv5," *Micromachines,* vol. 12, p. 1273, 2021.
[41] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658-666.
[42] X. Zeng, "One stage fine grained classification," Master Thesis Report,kth royal institute of technology 2021.
[43] H. Zhang, M. Tian, G. Shao, J. Cheng, and J. Liu, "Target detection of forward-looking sonar image based on improved yolov5," *IEEE Access,* vol. 10, pp. 18023-18034, 2022.
[44] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics,* vol. 10, p. 279, 2021.
[45] N. Al-Qubaydhi, A. Alenezi, T. Alanazi, A. Senyor, N. Alanezi, B. Alotaibi, M. Alotaibi, A. Razaque, A. A. Abdelhamid, and A. Alotaibi, "Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning," *Electronics,* vol. 11, p. 2669, 2022.
[46] X. Lang, Z. Ren, D. Wan, Y. Zhang, and S. Shu, "MR-YOLO: An Improved YOLOv5 Network for Detecting Magnetic Ring Surface Defects," *Sensors,* vol. 22, p. 9897, 2022.
[47] C.-M. Yan and C. Wang, "Automatic Detection and Localization of Pulmonary Nodules in CT Images Based on YOLOv5," *Journal of Computers,* vol. 33, pp. 113-123, 2022.
[48] M. Lamane, M. Tabaa, and A. Klilou, "Classification of targets detected by mmWave radar using YOLOv5," *Procedia Computer Science,* vol. 203, pp. 426-431, 2022.

[49] P. Tripathi, P. Chaudhary, and V. Kumari Ritika, "SYSTEM FOR DETECTING POTENTIAL WEAPON THREAT ON SURVEILLANCE," *Journal of Pharmaceutical Negative Results,* pp. 6001-6011, 2022.

[50] M. M. Fernandez-Carrobles, O. Deniz, and F. Maroto, "Gun and knife detection based on faster R-CNN for video surveillance," in *Pattern Recognition and Image Analysis: 9th Iberian Conference, IbPRIA 2019, Madrid, Spain, July 1–4, 2019, Proceedings, Part II*, 2019, pp. 441-452.

[51] A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning," *Neurocomputing,* vol. 330, pp. 151-161, 2019.

[52] S. Ahmed, M. T. Bhatti, M. G. Khan, B. Lövström, and M. Shahid, "Development and optimization of deep learning models for weapon detection in surveillance videos," *Applied Sciences,* vol. 12, p. 5772, 2022.

[53] M. A. Fatoumi, M. I. Kalache, and A. Krobba, "Étude et amélioration des performances de signaux GNSS," *Mémoire de Master, Université Medea,* 2022.

[54] G. Georges, "Algorithmes de calcul de positions GNSS basés sur les méthodes des moindres carrés avancées," Université de Technologie de Belfort-Montbeliard, 2016.

[55] S. T. S. Dr. R Guru, Thryambak M V, Shylesh N, V Hemanth, "Real time ship detection using YOLOv5," *International Research Journal of Engineering and Technology (IRJET),* vol. 09, 2022

[56] A. Mariscal-Torres, L. Ortega-Máynez, and J. Mejia, "Intelligent Surveillance Systems: A Review," *Cultura Científica y Tecnológica,* vol. 17, 2020.

[57] S. W. Ibrahim, "A comprehensive review on intelligent surveillance systems," *Communications in science and technology,* vol. 1, 2016.

[58] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proceedings-Vision, Image and Signal Processing,* vol. 152, pp. 192-204, 2005.

[59] L. Y. FU, "VIDEO SURVEILLANCE USING DEEP LEARNINGWITHFEW DATA SAMPLES," Final Year Project Report,UNIVERSITI TUNKU ABDUL RAHMAN, 2020.

[60] E. Jose, M. Greeshma, M. T. Haridas, and M. Supriya, "Face recognition based surveillance system using facenet and mtcnn on jetson tx2," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2019, pp. 608-613.

Chapter IV

Presentation and Realization of the Project

IV.1. Introduction

In the previous chapters we have worked on the theoretical side of the project, on the other hand in this last chapter we work on the hardware side. Indeed, in this chapter we will describe the components used and their functions in our system which designed to detect and alert the user of potential security threats in real-time, providing a more efficient and reliable method of monitoring and securing a specific area. Design hardware and software of our work will be presented and explained, as well as the various tests carried out. This chapter presents the prototyping of our security system using artificial intelligence methods, and wireless communication technologies such as GSM and radio frequency. By implementing this system, it is expected to provide a more comprehensive and effective system to security monitoring, addressing the limitations of traditional surveillance systems.

IV.2. Software and hardware components

The realization of this project requires knowledge of the equipment and integrated circuits used for the total functionality of the security system. Therefore, the objective of this part is to give the description of the hardware components that are needed. Moreover, three blocs divide the components used in this project: Processing units, Arduino shields, and Wireless communication.

IV.2.1. Processing Units

Within our project, the processing units play a fundamental role in the efficient and accurate operation of the security system using artificial intelligence methods and wireless transmission technologies. These units serve as the computing backbone and are responsible for processing and analyzing data collected from various sensors and cameras.

Processing units, which can include powerful microcontrollers, single-board computers, or dedicated AI processors like NVIDIA Jetson Nano, provide the computing power and resources needed to run complex algorithms and AI models. They enable real-time data processing, enabling rapid decision-making and response to security threats.

By integrating methods of artificial intelligence, the processing units are responsible for executing sophisticated algorithms such as object recognition, face recognition and anomaly detection. These algorithms enable the system to identify and classify security-related events to ensure detection of potential threats and provide enhanced security measures.

IV.2.1.1. Arduino boards

Arduino is a software company, it is an open-source consumer community, manufacturing single-board microcontrollers that are used to create interactive modules and digital devices that can detect and control things in real world.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs, a 16MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Arduino code is written in C++ programming language. It will interact with buttons, LEDs, motors, speakers, GPS (Global Positioning System) units, cameras, the internet, and even your smart-phone. Arduino consists of both a physical programmable circuit board and a piece of software, or IDE (Integrated Development Environment).

It contains:

- Microcontroller: ATmega328.
- Operating Voltage: 5V.
- Input Voltage (recommended): 7-12V.
- Input Voltage (limits): 6-20V.
- Digital I/O Pins: 14 (of which 6 provide PWM output).
- Analog Input Pins: 6.
- DC Current for 3.3V Pin: 50 mA.
- Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by boot loader.
- SRAM: 2 KB (ATmega328).
- EEPROM: 1 KB (ATmega328).
- Clock Speed: 16 MHz.

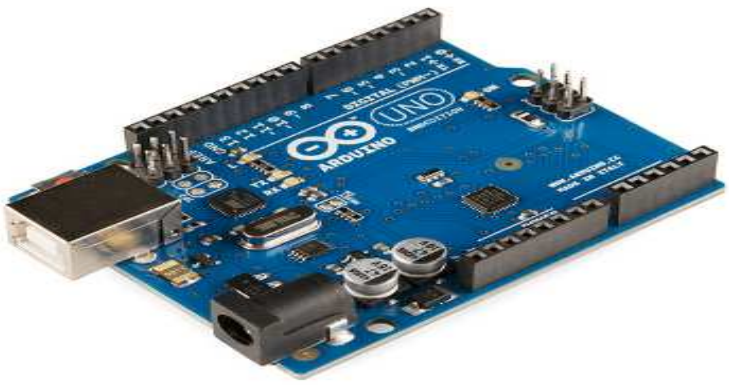


Figure IV. 1. Arduino Uno boards.

Another type of Arduino, called the Arduino Nano, was released in 2008. Moreover, it is one of the most popular Arduino boards available. Arduino Nano boards are equipped with the same microcontroller available in the Arduino Uno board, Atmega328p. The only thing missing is a DC power jack and it works with a Mini-B USB cable rather than a standard one. It contains:

- Microcontroller: ATmega328.
- Operating voltage: 5 V.
- Input voltage (VIN) : 6-20 V.
- Power consumption: 19 mA.
- SRAM: 2 KB.
- EEPROM: 1 KB.
- Flash memory: 32 KB, of which the boot loader uses 2 KB.
- DC current per I/O pin: 40 mA (20 mA recommended).
- Digital I/O pins: 22.
- Weight: 7 g.

Figure IV. 2. Arduino Nano boards.

IV.2.1.2. Nvidia Jetson Nano Card

The Jetson Nano Developer Kit is a small but powerful edge device from Nvidia. As one of the latest offerings from Nvidia, Jetson Nano features a GPU coprocessor based on the Maxwell micro architecture (GM20B). It comes with one streaming multiprocessor (SM) with 128 cores that can be used for parallel workloads including neural network calculations.

Jetson Nano holds an on-board 4 GB of LPDDR4 DRAM which is shared by the CPU (Quad-core ARM Cortex-A57) and the GPU accelerator [1], small size (69.6 × 45 mm). It is compatible with the most popular artificial intelligence frameworks: TensorFlow, PyTorch, Caffe, Keras, MXNet, etc. In addition, the Isaac SimSim package (included in the Isaac SDK by NVIDIA) is intended to provide a training environment for autonomous machines.

The table below indicates the Technical specifications of the NVIDIA Jetson Nano B01card.

Table IV. 1. NVIDIA Jetson Nano Card Technical Specifications.

GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (not included)
Video Encoder	4K @ 30 | 4x 1080p @ 30 | 9x 720p @ 30 (HEVC)
Video Decoder	4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 18x 720p @ 30| (HEVC)
Camera	2x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI 2.0 and eDP 1.4
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I2C, I2S, SPI, UART
Mechanical	100 mm x 80 mm x 29 mm

Figure IV. 3. NVIDIA Jetson NanoB01.

IV.2.1.3.1. Terminal

Since the NVIDIA Jetson Nano image is built from Ubuntu, we can use Terminal for administrative tasks such as creating/editing files and folders or compiling and executing programs. We can find the NVIDIA Jetson Terminal by clicking Search at the top left. We have to type "Terminal" to see the terminal application. After clicking on the Terminal icon, we will get the Terminal application as seen in (Figure IV.4) [2, 3].

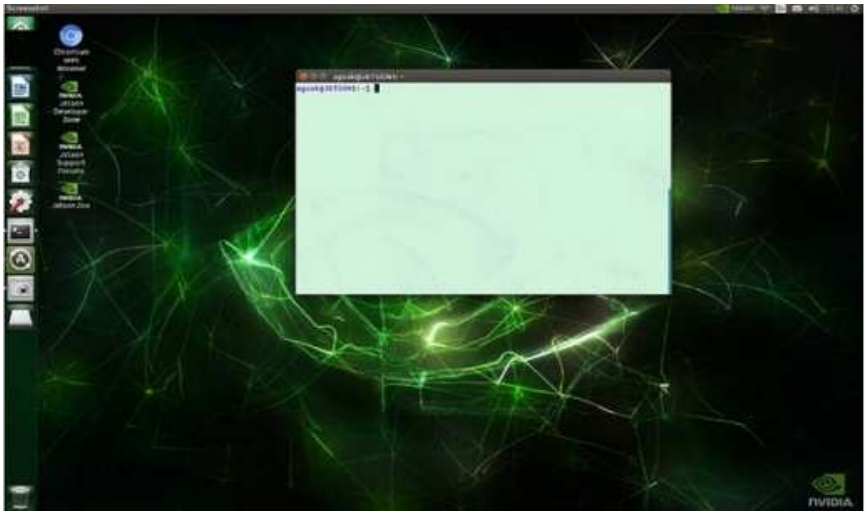


Figure IV. 4. Terminal Application.

IV.2.1.3.2. Remote Access to the NVIDIA Jetson Nano

The NVIDIA Jetson Nano device has a built-in network module with Ethernet. We can plug a LAN cable into the Ethernet port on an NVIDIA Jetson Nano. (Figure IV.5) **S**hows my NVIDIA Jetson Nano device with a plugged-in UTP LAN cable. Once the NVIDIA Jetson Nano device is connected to a network, we can verify its IP address using the ifconfig command in Terminal. Ifconfig you will see the IP address of your NVIDIA Jetson Nano.

Figure IV. 5. Connecting a UTP LAN cable to a NVIDIA Jetson Nano.

If we do not see the IP address, our network probably does not have a DHCP server. However, we can configure a static IP address on the NVIDIA Jetson Nano using terminal. We can modify a file in /etc/ network/interfaces. We will use the nano program. If our NVIDIA Jetson Nano does not have this program, we can install it as follows:

- **sudo apt-get install nano**

Now, we can modify a file in /etc/network/interfaces using nano, as follows:

- **sudo nano /etc/network/interfaces**

Then, we can write a static IP address. For instance, if we wanted to set our IP address to be 192.168.1.10 and gateway IP address to be 192.168.1.1, we could write these scripts in the /etc/network/interfaces file:

iface eth0 inet static address 192.168.1.10

netmask 255.255.255.0

gateway 192.168.1.1

We save the file. Now NVIDIA Jetson Nano has a static IP address [2].

IV.2.2. Arduino shields

Arduino shields play a significant role in our project, contributing to the functionality and expandability of the Arduino platform. These shields are additional hardware modules that can be easily connected to an Arduino board, providing specific functionalities and features that are essential for our security system.

The Arduino shields act as specialized add-on boards, extending the capabilities of the Arduino platform to meet the specific requirements of our project. They provide additional input/output interfaces, communication protocols, sensors, actuators, or other functionalities that are necessary for the proper functioning of our security system.

IV.2.2.1. LCD display

LCD (Liquid Crystal Display) screen is an electronic display module and used in a wide range of applications including computer monitors, televisions, instrument panels. It is easy to interface with a microcontroller because of an embedded controller. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines.

This LCD has two registers, namely, Command and Data. The command instructions given to the LCD to carry out a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. and it stored in the Command register. A command is an instruction given to LCD to do the data that will display in on the LCD is kept in the data register.

Figure IV. 6. 16×2 LCD display.

As an example of programming an LCD:

```
#include<LiquidCrystal.h>
LiquidCrystallcd (2, 3, 4, 5, 6, 7);
```

IV.2.2.2. Buzzer

A buzzer or beeper is an audio signaling device for Arduino, which may be mechanical, electromechanical, or piezoelectric. The operating voltage is from 3.3V to 5V. Alarm devices and timers are two common applications for buzzer.

Figure IV. 7. Buzzer.

IV.2.3. Wireless communication

Wireless communication technology eliminates the need for physical cabling and allows flexibility in the placement and deployment of devices within our security system. It allows for ease of installation and scalability as devices can be easily added or moved without the

limitations of wired connections. It allows the transmission of data between the different elements of our security system. It also allows for the transmission of security event information, status updates, sensor readings and control commands, ensuring efficient communication and coordination between the various system components.

One of the key advantages of wireless communications is their ability to provide real-time monitoring and response. Security-related events or alarms can be immediately reported to the central control unit or monitoring station, allowing immediate action to be taken. This ensures timely detection of security threats and enables quick response and remedial action.

IV.2.3.1. Access point

An access point (AP) is a device that serves as a wireless communications hub, allowing wireless devices to connect to a wired network. Creates a local area network (LAN) by broadcasting a Wi-Fi signal that devices can detect and connect to. Access points play a crucial role in enabling wireless connectivity in homes, offices and public spaces.

Figure IV. 8. Access Point.

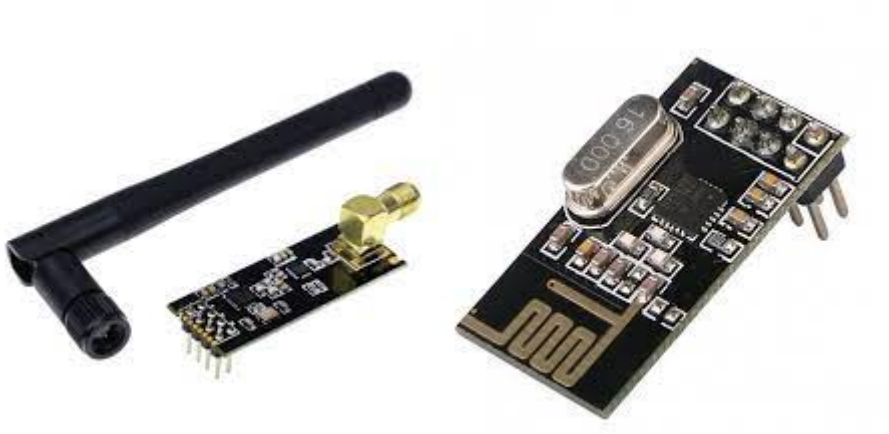
In our project, the access point is establishing and managing wireless connections with the devices in our security system. It acts as a base station that broadcasts a Wi-Fi signal and allows devices like the IP camera, laptop, Smartphone, and Nvidia Jetson Nano to connect wirelessly. Moreover, when a device in our security system wants to communicate, it connects to the access point by connecting to the access point's Wi-Fi network that assigns an IP address to the connected device, so it serves as a unique identifier in the network. This addressing allows devices to send and receive packets of data to and from other devices within the network. In addition, access points play a role in managing network security, in which we can implement security measures such as encryption and authentication to ensure

only authorized devices can connect to the network. This helps protect our security system from unauthorized access and data breaches.

In addition, the access point can support features such as quality of service (QoS) and traffic prioritization. This allows us to effectively allocate bandwidth resources and ensure that critical data such as security alerts or real-time video streams take precedence over less time-sensitive traffic. This helps maintain the performance and responsiveness of our security system.

IV.2.3.2. nRF24L01 Modules

This nRF24l01 is a wireless remote module, which means this module can both send and receive information. It uses the 2.4 GHz band, which falls under the ISM band, and hence it is legal to use in almost all countries for engineering applications. And it can operate with baud rates from 250 kbps up to 2 Mbps. The module can use 125 different channels, which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time. The module operates at 3.3V hence can be easily used with 3.2V systems or 5V systems we can easily connect it to an Arduino without using any logic level converters. The modules when operated efficiently can cover a distance of 100 meters (200 feet) which makes it a great choice for all wireless remote controlled projects.



(a) (b)

Figure IV. 9. (a) nRF24L01-PA-LNAModule, (b) nRF24L01 Module.

IV.2.3.2.1. Role of the nRF24L01 Modules in the Project

These modules are transceiver, which means they can transmit and receive a radio frequency of 2.4 GHz. The nRF24L01 module is used to transmit information between the Arduino Uno

and the GSM in the transmitter section and transmit information between the Arduino Uno and the LCD display in receiver section.

In the transmitter section, the RF technology is employed to convert the digital information from the security system into radio waves for wireless transmission. The transmitter processes the data, modulates it onto an RF carrier signal, and amplifies it to an appropriate power level for transmission. Moreover, in the receiver section, the RF technology is utilized to capture the radio waves carrying the transmitted data. The receiver module receives the RF signal, demodulates it to recover the original digital information, and forwards it to the appropriate processing unit. RF receivers are designed to be sensitive to the desired radio frequency signal while rejecting interference and noise to ensure accurate data reception. The role of RF in both the transmitter and receiver is essential for wireless communication in the security system. It enables the transmission of data from the sensors and cameras to a central processing unit, providing real-time monitoring and detection of security threats.

The nRF24L01 plays a dual role as both a transmitter and receiver, and we can represent this functionality in the same figures. Figure IV.10 illustrate the connection diagram for the nRF24L01, while Figure IV.11 demonstrate the corresponding electrical assembly circuit.

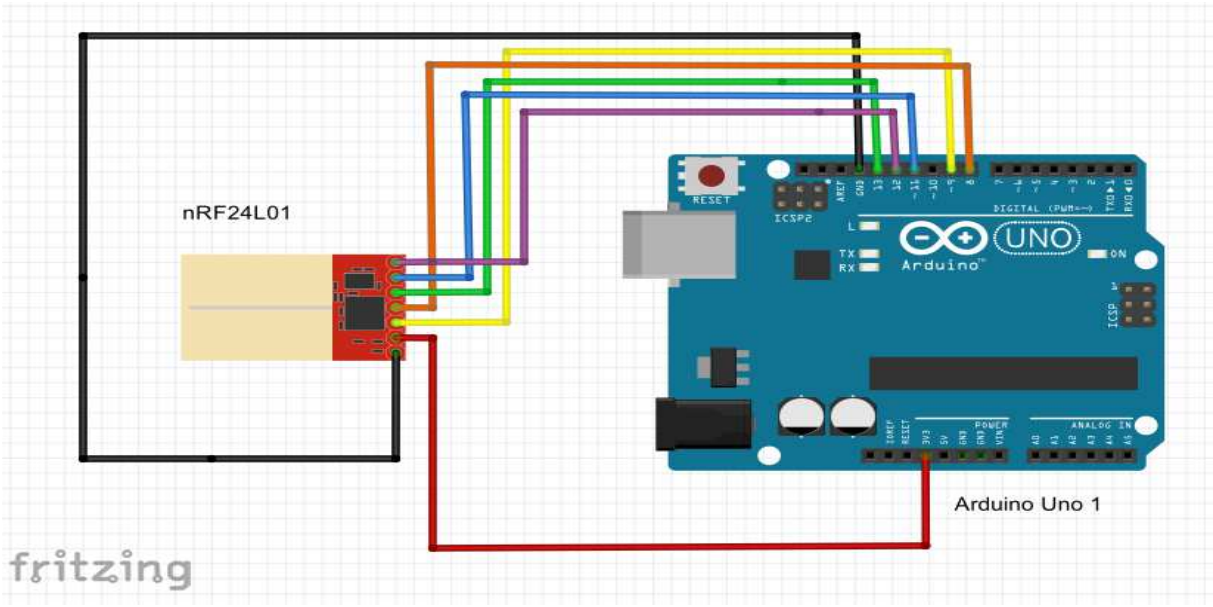


Figure IV. 10. nRF24L01 Radio frequency Transmitter/Receiver connection with the Arduino Uno board.

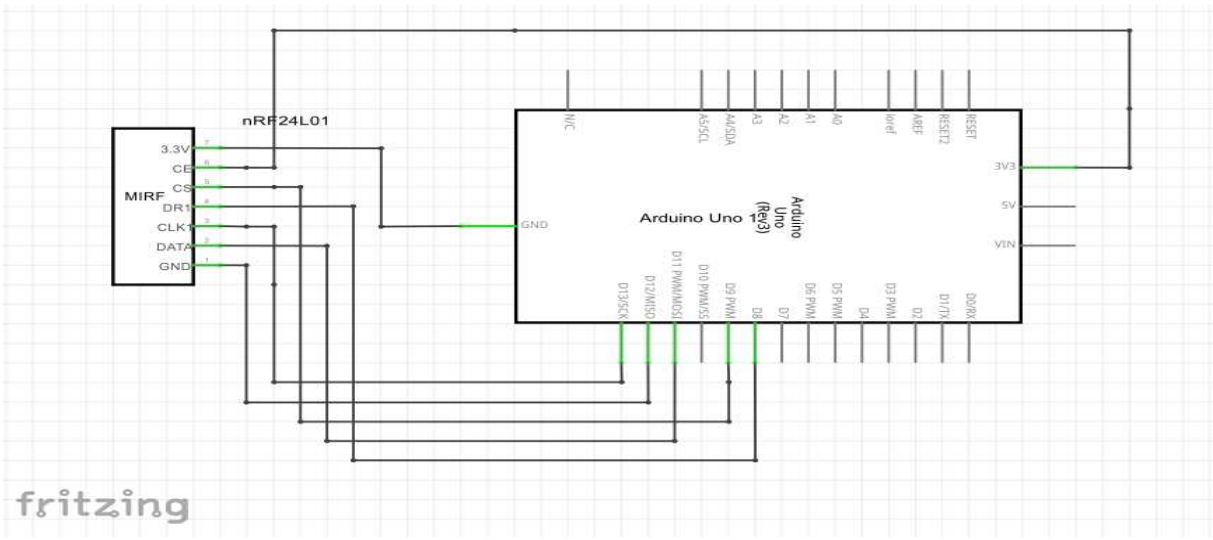


Figure IV. 11. Electrical circuit of nRF24L01 radio frequency Transmitter/Receiver Module with Arduino Uno Board.

To illustrate the programming of the nRF24L01 for the transmitter, let's consider the following example:

```
#include "SoftwareSerial.h"
#include <RF24.h>
#define Enable_A  8
#define pinCE   9
#define pinCSN  10
RF24 radio(pinCE, pinCSN);
const byte adresse[5] = {'R', 'x', 'A', 'A', 'A'};
char message[1];
char input = 0;
void setup() {
radio.begin();
radio.openWritingPipe(adresse);
radio.setPALevel(RF24_PA_MIN);
radio.stopListening();
pinMode(Enable_A, OUTPUT);
Serial.begin(9600); // connect serial
}
```

Based on the provided information, if the channel configuration for the transmitter is {'R', 'x', 'A', 'A', 'A'}, the receiver would have the same channel configuration. This means that both the transmitter and receiver will be using the same channel settings to establish communication.

For the receiver, we use the following code as an example:

```
//#include <SoftwareSerial.h>
#include<LiquidCrystal.h>
LiquidCrystallcd(2, 3, 4, 5, 6, 7);
//#include <SPI.h>
#include <RF24.h>
#definepinCE   9 //The "CE" pin of the NRF24L01 is connected to the digital output D7 of the Arduino.
#definepinCSN 10//The CSN pin of the NRF24L01 is connected to the digital output D8 of the
//Arduino.
#define led_pin1 8
RF24 radio(pinCE, pinCSN);    //  NRF24L01
const byte adresse[5] = {'R', 'x', 'A', 'A', 'A'};
char message[1];
char msg1;
void setup() {
// Initialisation of the serial port (to display received information on the Arduino IDE Serial Monitor)
Serial.begin(9600);
Serial.println("Récepteur NRF24L01");
Serial.println("");
```

IV.2.3.3. GSM Module

SIM900 GSM/GPRS shield is the GSM module used in our system. The shield is designed to operate at 3.3V or 5V and works easily with, and connects directly to the Microcontroller. It can be connected to the Arduino by using a UART connection. The next Figure shows the GSM/GPRS Shield.

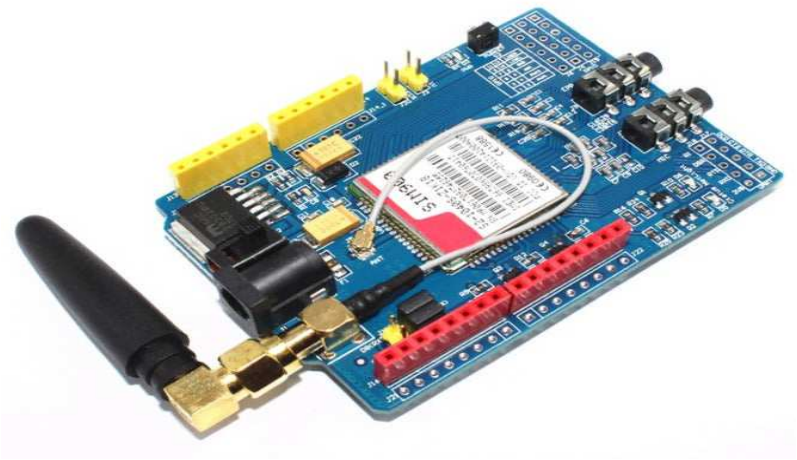


Figure IV. 12. GSM/GPRS Shield.

- **Features:**
  - ✓ Compatible with Arduino and clones.
  - ✓ Quad band support : 850/900/1800/1900MHz.
  - ✓ Operation temperature: -40°C to +85 °C.
  - ✓ Full control via AT commands set: Standard - GSM 07.07 & 07.05 and Enhanced - SIMCOM AT Commands.
  - ✓ Support TCP/UDP protocols.
  - ✓ Low power consumption: 1.5mA (sleep mode).
  - ✓ Dimension : 71mm x 66mm x 12mm.

There are several Arduino GSM/GPRS Shields of different frequency bands allowing communication with the whole world, whatever the type of communication (SMS, voice, data and fax). In this project, we have chosen the GSM SIM900. Gather 850/900/1800/1900MHz, it can transmit voice, SMS and phone information data with low power consumption. It allows you to exchange messages with the Arduino Uno board using AT commands.

IV.2.3.3.1. Connecting the GSM Module

The SIM900 module is placed directly on the Arduino board (see Figures IV.13 and IV.14). A voltage of 5V powers the Arduino Uno board and the SIM900. GSM module communicates with the Arduino board by AT commands (see command Figure IV.15) which allow the exchange of SMS or calls between the card and the GSM.

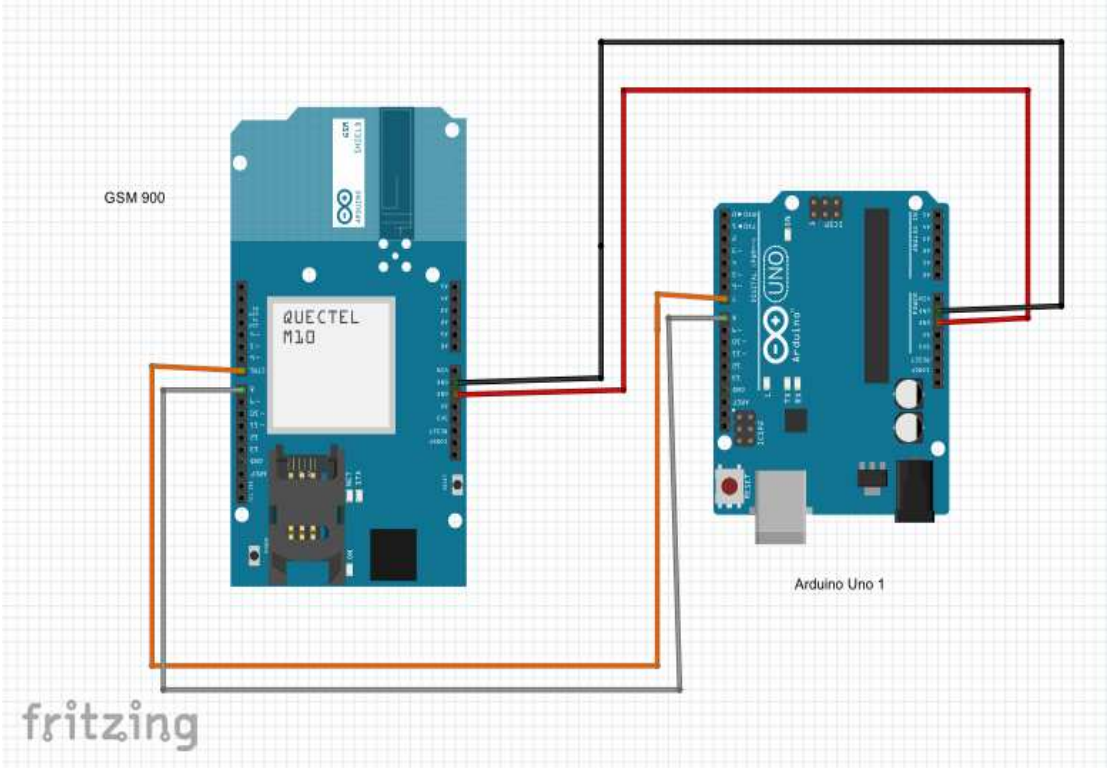


Figure IV. 13. Connecting GMS Module with Arduino Uno.

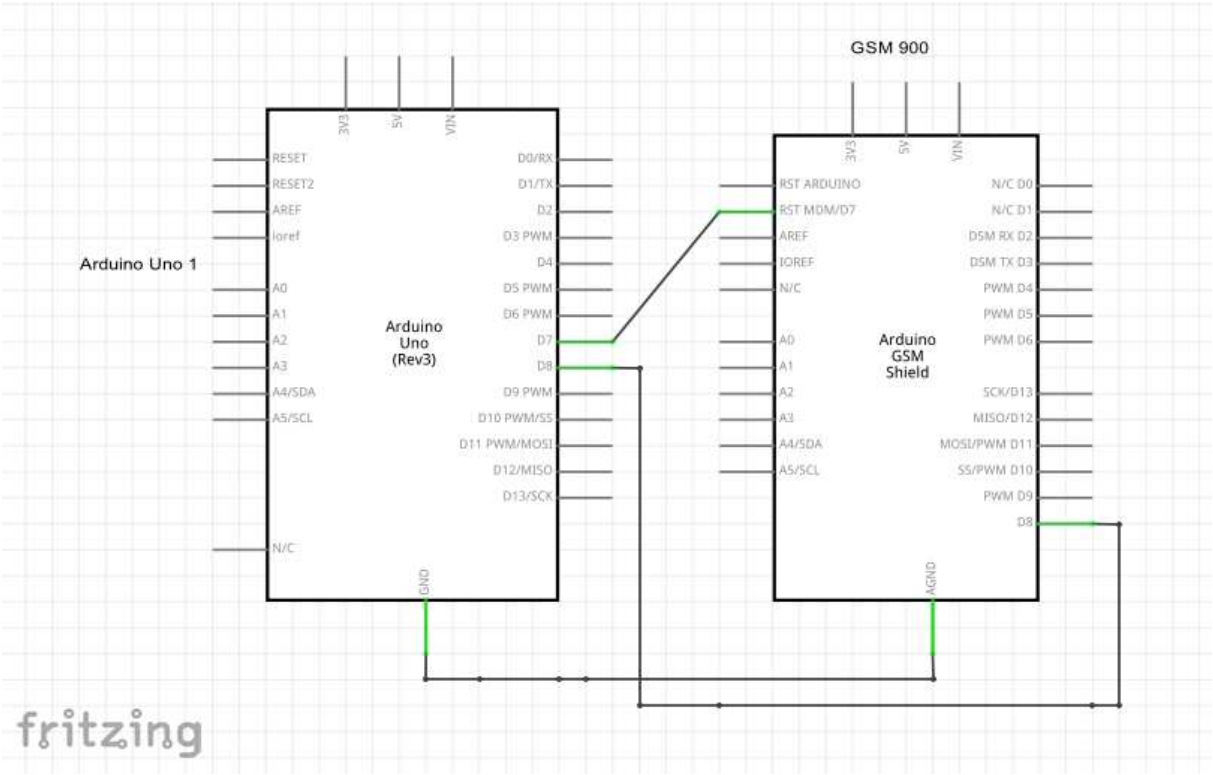


Figure IV. 14. Electrical Circuit of the GSM Module with the Arduino Uno.

As an example of programming a GSM:

```
#include <SoftwareSerial.h>;
SoftwareSerial SIM900(8, 7);
const intinputA   = 9;
int D;
char incoming_char = 0;
intsalir = 0;
void setup() {
SIM900.begin(19200);
delay(25000);
Serial.begin(19200);
Serial.println("OK");
pinMode(inputA, INPUT);
}
void loop() {
  D = digitalRead(inputA);
if (D == 1)  {
call(); //call
    //  message_sms(); // Send a message
  }
}
```

The different functions in Arduino programming are illustrated in the following:

```
Void message_sms() {
SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
delay(100);
SIM900.println("AT+CMGS=\"+213**********\""); // recipient's mobile number, in
international format
delay(100);
SIM900.println("Alarm"); // message to send
delay(100);
SIM900.println((char)26);
delay(100);
SIM900.println();
delay(5000);
Serial.println("SMS sent successfully");
}
```

IV.2.3.3.2. AT Commands

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. The GSM 07.07 specifies the AT command set for control Mobile Equipment (ME) or Data

Terminal Equipment (DTE), such as a MCU from a Terminal Equipment (TE) through Terminal Adaptor (TA) or Data Circuit Terminating Equipment (DCE) [4].

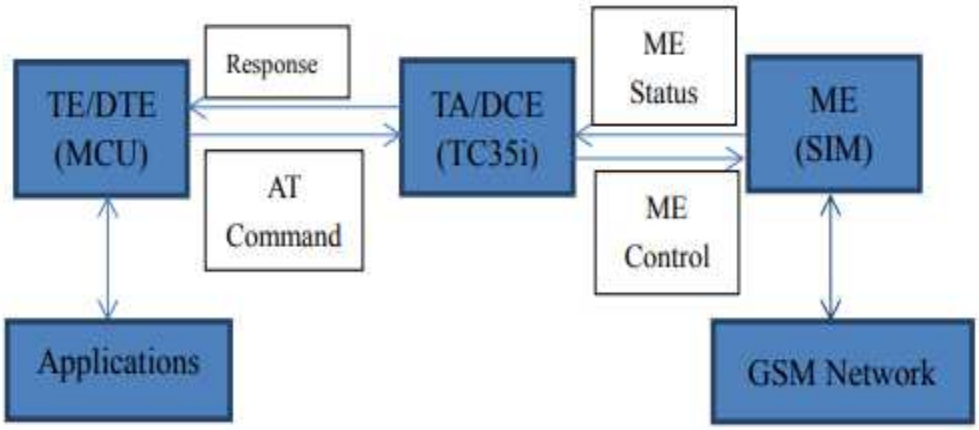


Figure IV. 15. The operation link between TE, TA, and ME.

IV.2.3.3.3. Role of the GSM Module in the Project

The GSM module plays a vital role in our smart security system by sending coordinates through SMS or calls to notify the security center about various emergencies detected in the environment we are securing. This enables the security agents to promptly respond to the situation, ensuring rapid intervention and effective control.

When our system detect an emergency without intervention of human, the GSM module initiates the communication process. It sends the precise coordinates of the incident to the security center using SMS or by making a call. This immediate notification allows the security agents to be promptly informed about the location of the emergency. By receiving the coordinates, the security center can quickly determine the exact position where the emergency is occurring. This information enables the security agents to swiftly mobilize and reach the location of the incident, facilitating rapid intervention.

IV.2.3.4. Camera IP Imou Bullet 2E

An IP camera (Internet Protocol Camera) is a type of security camera that uses the Internet Protocol (IP) and has its own IP address. IP cameras allow you to monitor specific areas from any remote location, in contrast to analog CCTV cameras that transmit the video images via cables with a limited length. In our project. we will used the IMOU Bullet 2E 2MP.

Figure IV. 16. Camera Imou Bullet 2E.

The IMOU Bullet 2E 2MP external security camera is a fixed outdoor camera that features:

- Extremely High Resolution (both day and night).
- Maximum memory card size 256 GB.
- Smart Colour Night Vision (full colour pictures, even in complete darkness).
- Resolution 2MP (1920 x 1080) Full HD.
- Diversified Storage.
- Night vision: up to 30m away.
- Network: 100Mbps Ethernet connection.
- Wi-Fi: 2,4 GHz (no 5,8 GHz).
- Human Detection.
- Built-in Spotlights, Microphone and Wi-Fi Hotspot.
- IP protection.

In our project, IP cameras have the role of capturing and transmitting visual data over the IP network, and enabling seamless integration with other security system components. In addition, they are an essential part for real time monitoring which providing visual feedback on the monitored area.

IV.3. Smart Security System

This section provides an overview of the overall architecture of the security system. It discusses the components, modules, and their interactions within the system.

The architecture as shown in (see Figure IV.17) is designed to ensure the effective integration of artificial intelligence methods and wireless transmission technologies to achieve real-time monitoring and security threat detection.

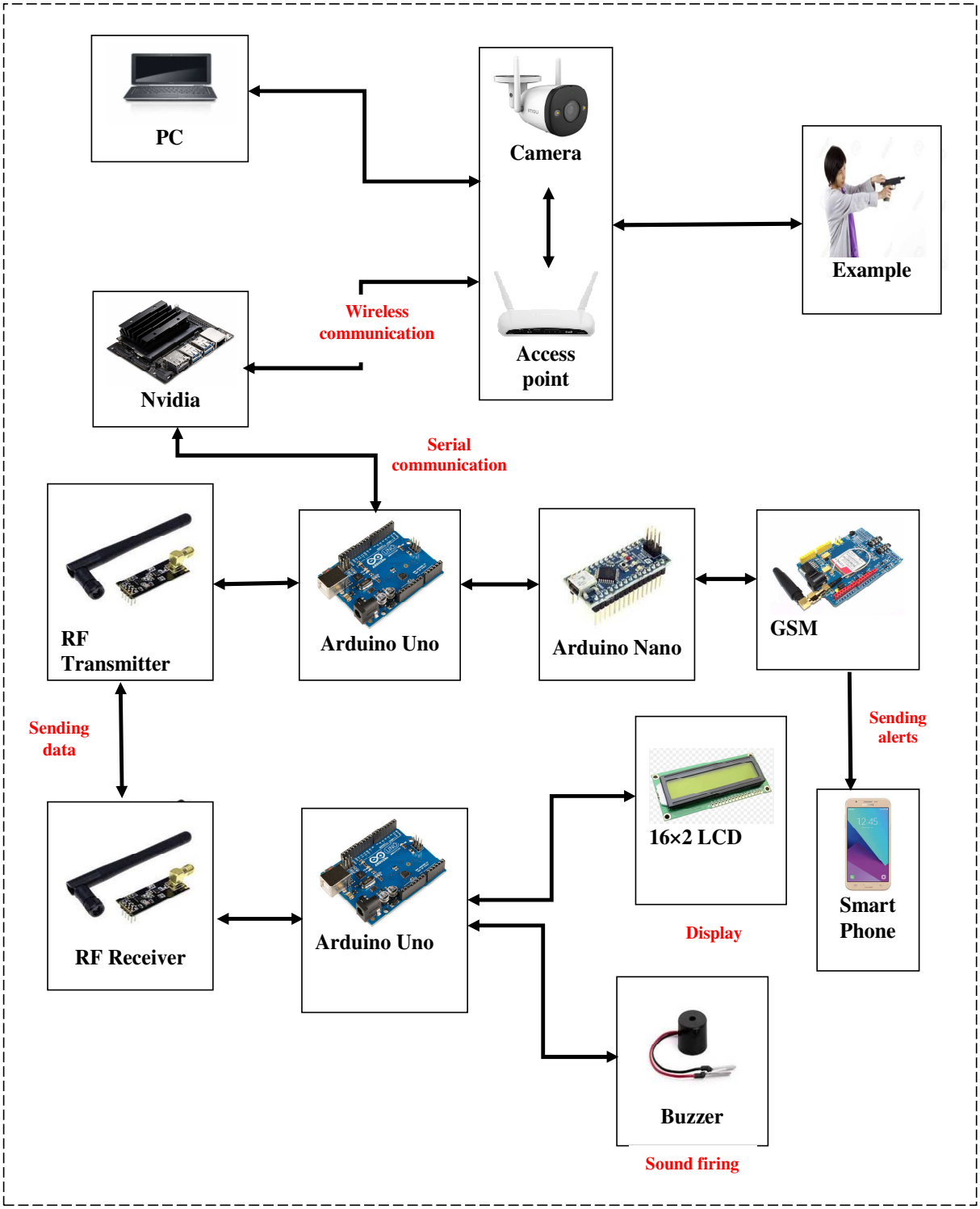


Figure IV. 17. System Design Architecture and Connection.

Our project is divided into two major parts: Offline and Online. The offline part focuses on simulation, while the online part focuses on realization and practical implementation.

For the part of simulation, we have presented it well systematically in chapter III and for the realization; it will be presented in this chapter. Without forgetting that our work has three deferent blocs:

- Sender part,
- Receiver part,
- Video processing.

So in order to realize the prototype of our monitoring and security system, it was important to have all the components in section IV.2. For this system, we need a radio frequency transmitter, a radio frequency receiver, a GSM system, and Nvidia Jetson Nano part for the video processing.

The material realization is made in the first place, each module of our assembly is realized and tested separately, and assemblies are first built on "breadboards" or experimental setup. After experimenting and adopting them separately, we have grouped together and made on multi circuits (GSM module, radiofrequency receiving, and radio frequency transmission).

The system consists of two sections; the transmitter section and the receiver section with a part of video processing.

IV.3.1. Transmitter section

This section serves as a crucial component in the overall security system and is responsible for capturing and transmitting video from surveillance cameras. Its main function is to collect real-time visual information from the environment and transmit it wirelessly to the receiver area for further processing and monitoring.

Also in this section, we address the hardware configuration, the video capture process, video processing techniques, wireless transmission technology, addressing and routing mechanisms, and the security measures used to ensure the integrity and confidentiality of the transmitted video data.

Effective implementation of the transmitter section gives the security system the ability to capture high-quality video images and transmit them seamlessly over a wireless network. This enables remote monitoring, real-time analysis and quick response to potential threats or security incidents. In the following sections, the various aspects of the transmitter area will be examined in detail and its essential role in providing a robust and efficient security system is highlighted.

IV.3.1.1. Hardware Setup

The transmitter section of our security system comprises several key components, each playing a specific role in the overall functionality. The components included in the transmitter section are as follows: Power supply, RF transmitter, GSM900 module, Arduino Nano,

Arduino Uno, Nvidia Jetson Nano B01, and an IP camera. These components are integrated into the experimental prototype, as illustrated in Figure IV.18.

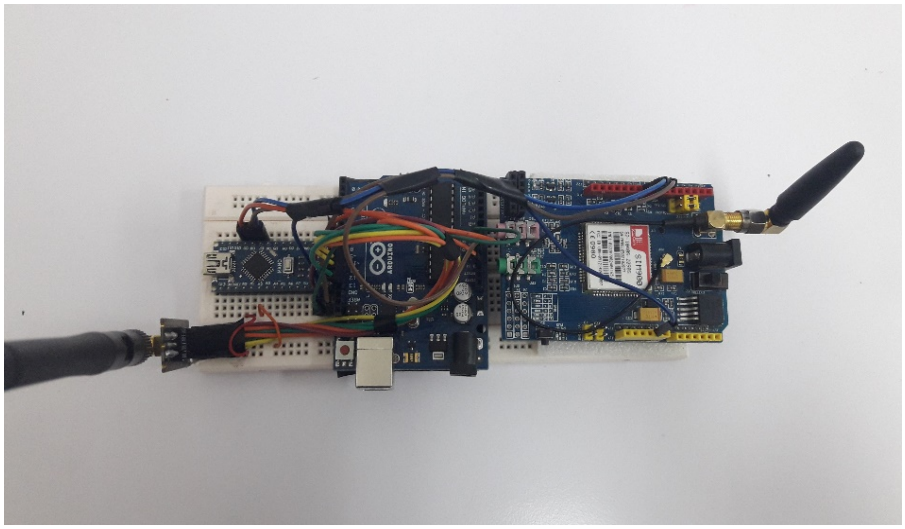


Figure IV. 18. Transmitter Prototype.

To enable the capture and transmission of video data, we have therefore implemented a comprehensive hardware configuration in which the following components play a crucial role:

- Power Supply: Used to provide adequate power to all components in the transmitter area. It guarantees uninterrupted operation and enables the system to function properly.

Figure IV. 19. Power Supply.

- RF Transmitter: this module (Figure IV.9.a) is built into the hardware configuration, enables wireless transmission of video data, also enables the captured video signal to be efficiently transmitted to the receiving area, ensuring real-time monitoring and analysis.

- GSM900: improves the capabilities of the transmitter part. Allows alternative communication options such as SMS notifications or calls. This module (Figure IV.12) provides additional flexibility and redundancy in the communication channels.
- Arduino Nano and Arduino Uno: serve as the brains of the transmitter section. They provide the necessary computing power and interface to various components. They are responsible for controlling the RF transmitter module, coordinating the flow of data, and managing the overall operation of the transmitter section. In addition, since the Arduino Uno (Figure IV.1) is not a multitasking component; we used the Arduino Nano (Figure IV.2).
- Nvidia Jetson Nano B01: It is a powerful single-board computer specially designed for artificial intelligence and deep learning applications. It is used in the transmitter part to process and analyze the video data in real time. Its high computing capacities enable advanced video analysis, object recognition and other AI-based functions, shown in (Figure IV.3).
- The IP camera: It is an integral part of the transmitter part and captures the video signal of the monitored environment. It is responsible for providing high-quality video footage, which is then processed and transmitted for further analysis and monitoring, shown in (see Figure IV.16).

The integration of these hardware components in the transmitter section creates a robust and efficient system for capturing, processing, and transmitting video data. The combined functionality of the power supply, RF transmitter, GSM900 module, Arduino boards, Nvidia Jetson Nano B01, and IP camera enables real-time video transmission, advanced analytics, and remote monitoring capabilities in the overall security system.

IV.3.1.2. Connecting components

Since us realized and tested separately each module of our assembly, and assemblies are first built on breadboards. After experimenting and adopting them separately, we have grouped them together.

Firstly, we linked the Power supply, RF Transmitter, GSM900, Arduino Nano, Arduino Uno, Nvidia Jetson Nano B01 and the IP Camera together with the PC (Figure IV.20). Which we have serial communication between:

- The PC and the IP Camera;
- Nvidia Jetson Nano and the Arduino Uno.

Figure IV. 20. Connecting Components of Transmitter Section.

However, for the Nvidia Jetson Nano and the IP Camera, we linked between them by an access point (Figure IV.21). In which we use some PC applications to detect the access point's Wi-Fi network for the IP Camera and the Nvidia Jetson Nano that assigns an IP addresses to the connected devices. We used "Configtool"(Figure IV.22) to detect the IP address of the IP Camera and "Advanced IP Scanner" (Figure IV.23) to detect the IP address of the Nvidia Jetson Nano after that we use "Remote Desktop Connection" to link the Nvidia Jetson Nano to the PC (Figure IV.24) and we get the IP addresses:

- Nvidia Jetson Nano IP address: 192.168.1.5
- IP Camera IP address: 192.168.1.2

This addressing allows devices to send and receive packets of data to and from other devices within the network.

Figure IV. 21. Connecting the Nvidia Jetson Nano and the IP Camera.

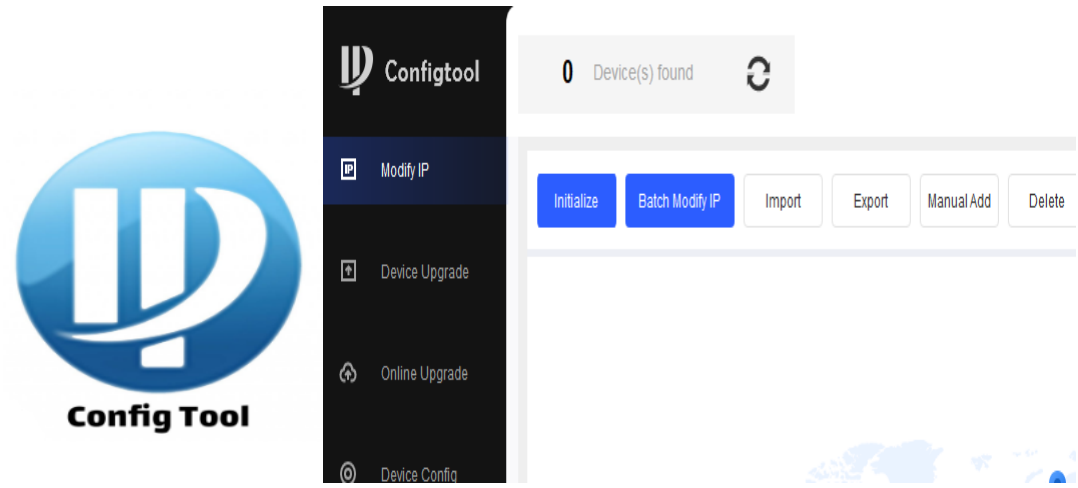


Figure IV. 22. Configtool.

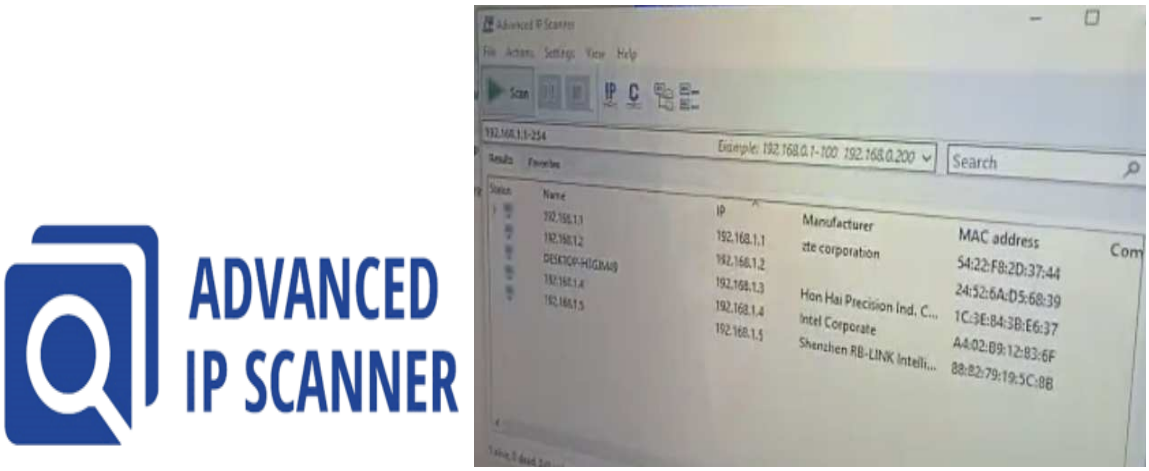


Figure IV. 23. Advanced IP Scanner.

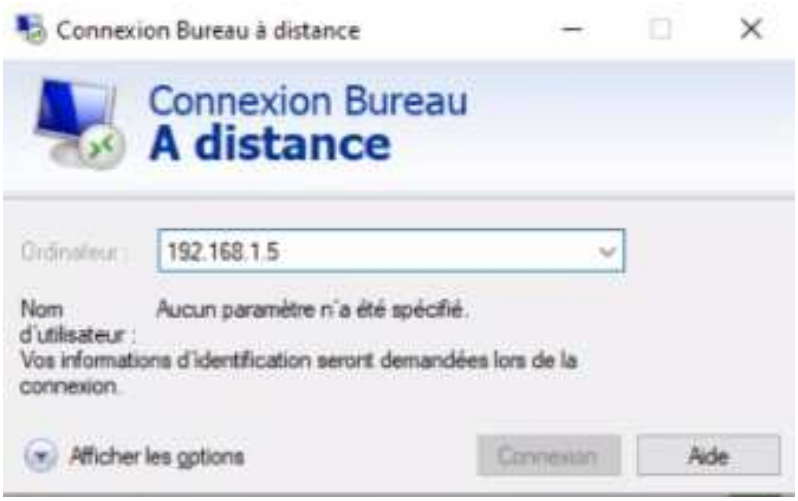


Figure IV. 24. Remote Desktop Connection to the Nvidia Jetson Nano.

Moreover, the rest of components: GSM900, the Arduino Nano, and the RF Transmitter are linked with Arduino Uno as shown in (Figure IV.25) below.

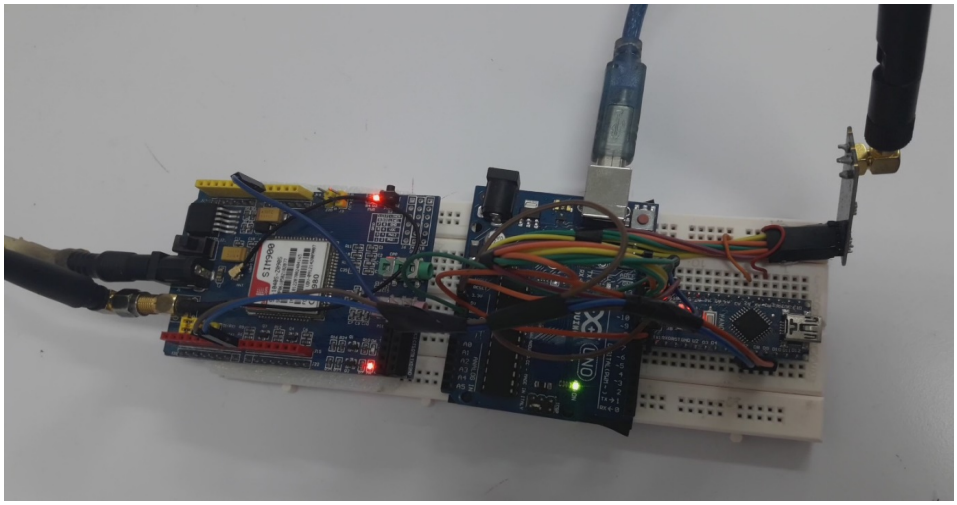


Figure IV. 25. ConnectingGSM900, Arduino Nano, and RF Transmitter with Arduino Uno.

Secondly, we activate each element of the transmitter section in which:

- We inter to the Nvidia Jetson Nano by using a PC (Figure IV.26).

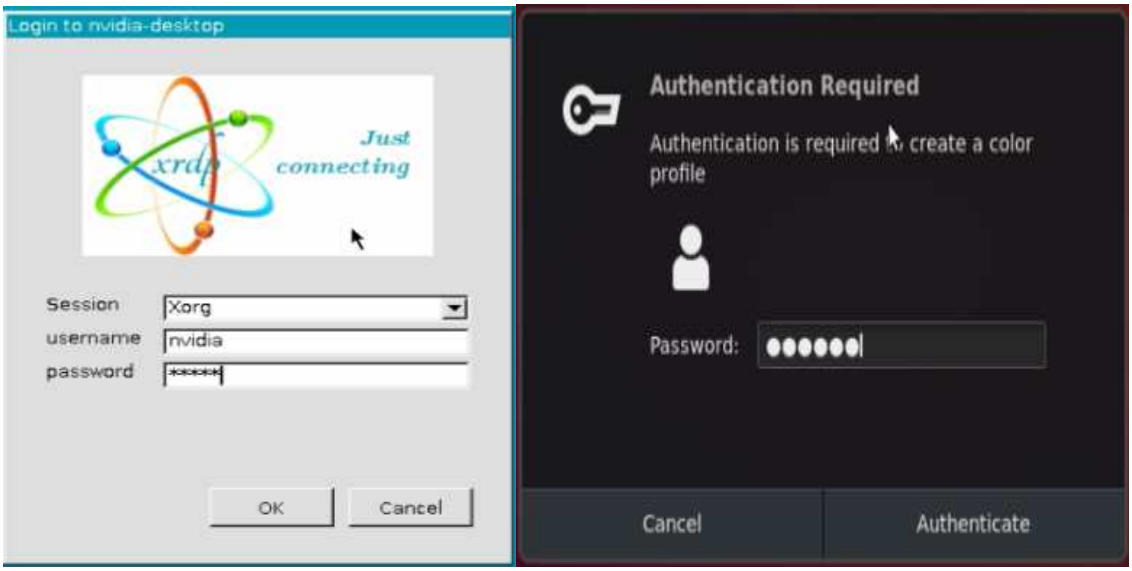


Figure IV. 26. Nvidia Jetson Nano Login.

Then we go to Files (Figure IV.27) and choose the file we worked on “Intelligent Monitoring System” in which we have added it by a USB flash to the Nvidia (Figure IV.28).

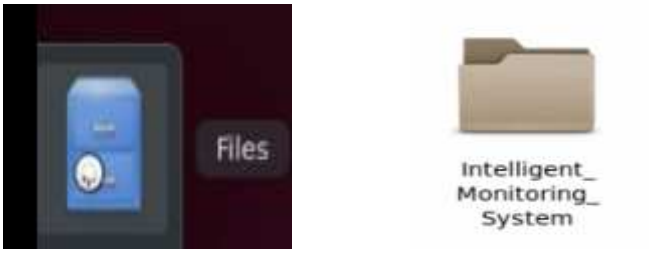


Figure IV. 27. Nvidia Jetson Nano, and Intelligent Monitoring System Files.

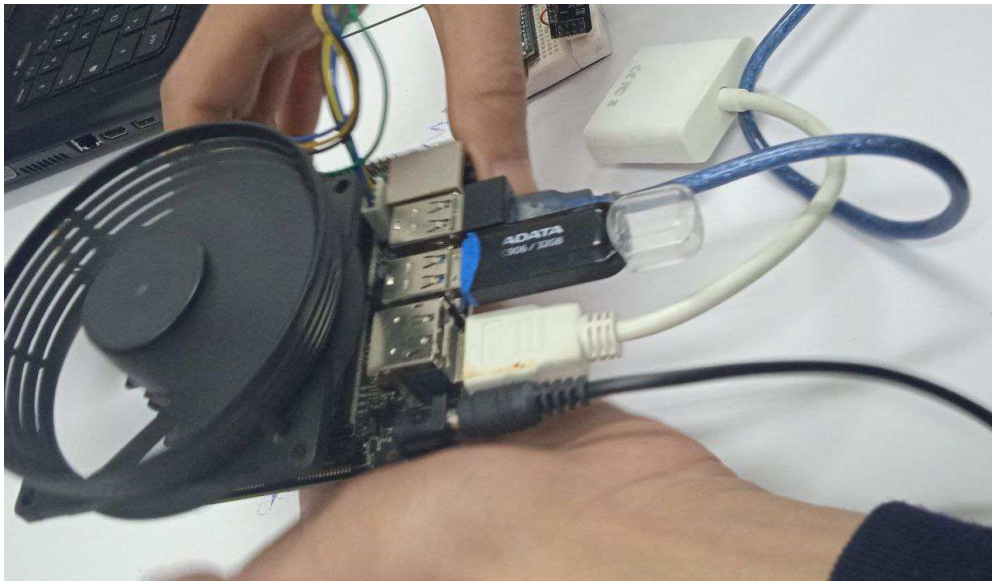


Figure IV. 28. Adding Files to Nvidia Jetson Nano by a USB flash.

Then we choose the folder "Sender" (Figure IV.29) to activate the nRF transmitter module to the Nvidia Jetson Nano.

Figure IV. 29. Adding sender code to Nvidia Jetson Nano.

Moreover, to make sure that Arduino Uno is working, we define the logical gates (Figure IV.30).

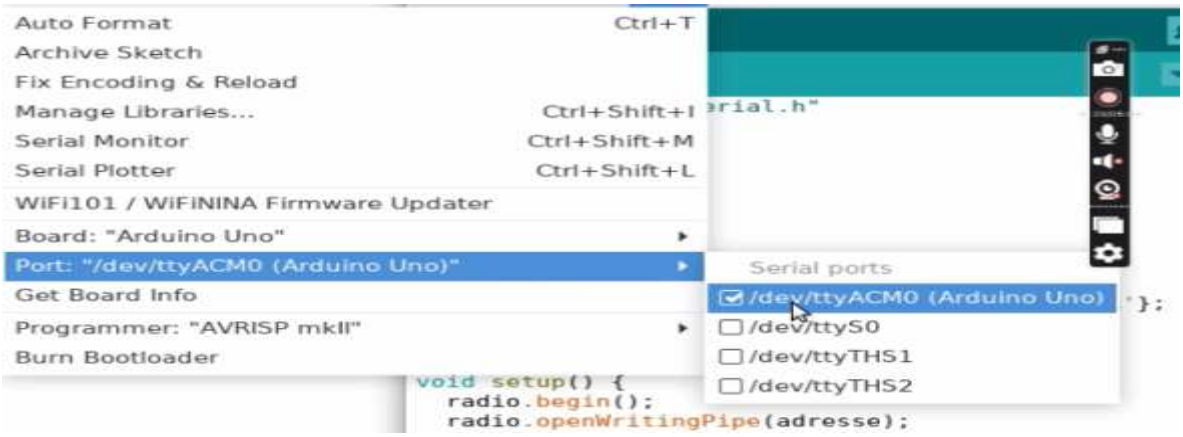


Figure IV. 30. (a) Define Arduino Uno and the Logical Gate.

Finally, after completing the activation of each components of the transmitter section, we do the same with the components of the receiver section to guarantee the access of data transmitted.

IV.3.2. Receiver Section

This section plays a crucial role in the overall security system by receiving and processing the video signals transmitted by the transmitter part. It serves as a central hub for receiving, decoding, and displaying video data captured by surveillance cameras.

Also in this section we focus on the hardware configuration, the process of receiving and decoding the video signal, the data processing techniques, and the display mechanism used in the receiver section. The receiver part is responsible for receiving the transmitted video data and converting it into a format that can be easily interpreted and viewed on connected devices. The RF receiver captures the radio signals and decodes them to extract the video information. Which means that the received video data is then processed using appropriate techniques to improve quality, perform analytics or apply any algorithms required for specific security applications.

The effective implementation of the receiver part gives the security system the ability to receive and display high-quality video streams in real time. This facilitates continuous monitoring and monitoring, enabling early detection and response to potential threats or security incidents.

The receiver part acts as an interface between the captured video data and the user, providing a visual representation of the monitored area. In the following sections, we delve into the various aspects of the receiver section, examining its hardware configuration, the signal reception and decoding process, video processing techniques, wireless communication

technology, addressing and routing mechanisms, and the security measures used to ensure the integrity and confidentiality of the received video data. By looking at these aspects holistically, we want to highlight the crucial role that the receptionist plays in establishing a strong and efficient security system.

IV.3.2.1. Hardware Setup

Receiver section consists: Power Supply, RF Receiver, Piezo Buzzer and LCD display as shown in the experimental prototype in Figure IV.31.

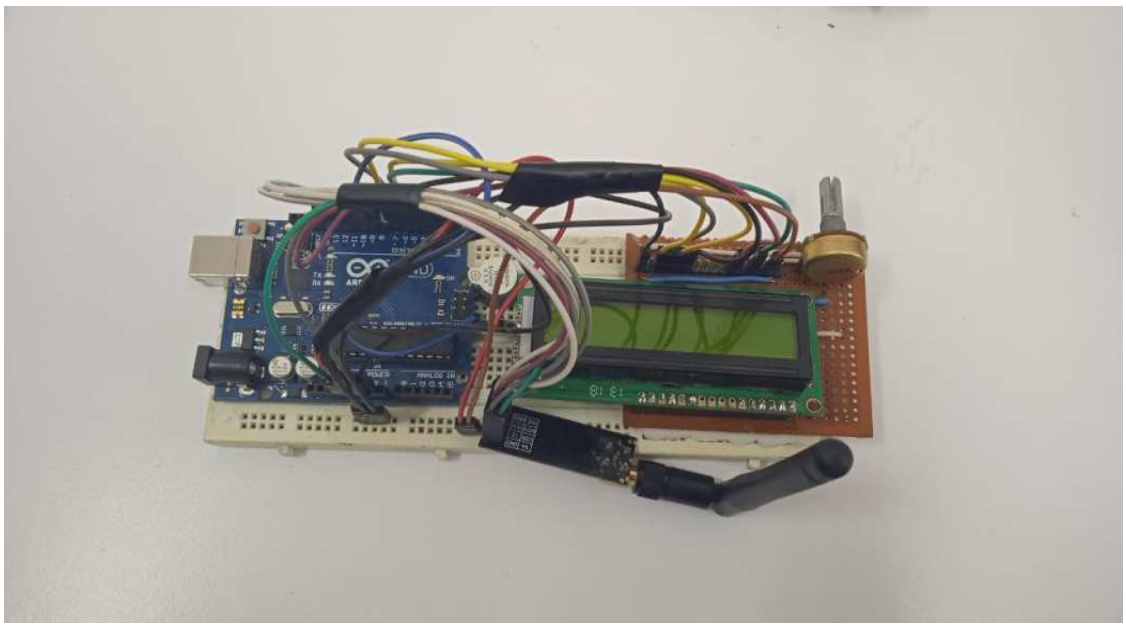


Figure IV. 31. Receiver Prototype.

The receiver section of our security system is made up of several essential components that work together to receive, process and display the transmitted video signals. In which the following components play a crucial role:

- RF Receiver: This module (Figure IV.9.b) is responsible for capturing the radio signals transmitted by the transmitter part. It receives the encoded video signals and converts them into a format that can be further processed and displayed. The RF receiver plays a crucial role in receiving video data reliably.
- Piezo Buzzer: is a small audio device (Figure IV.7) that generates audible alerts or notifications. In the receiver section it used to provide audio feedback or audible alarms in response to specific events or security triggers, and in our project it sends a voice alert when it receives that a person or weapon is detected by the security system.
- LCD Display: is a visual output component that displays received video data in a clear and user-friendly way. It serves as an interface between the user and the surveillance

system, allowing real-time monitoring and viewing of the captured video streams. The LCD screen provides a convenient and immediate visual representation of the area being monitored.

The hardware configuration in the receiver section ensures successful reception and display of the transmitted video signals. The power supply ensures stable operation while the RF receiver captures and decodes radio signals. The piezoelectric buzzer enhances the user's attention with audio feedback and the LCD screen presents received video data in a visual format.

The combination of these components enables efficient monitoring and analysis of captured video streams, enabling rapid detection and response to potential security threats. The receiver part with its hardware configuration plays a crucial role in ensuring the effective operation and functionality of the entire security system.

IV.3.2.2. Transmission Protocol

After sending the data from the transmitter section, the RF receiver will detect it and send it to the Arduino Uno that serves as a crucial component that enhances the functionality and control of the system. The Arduino Uno receives the decoded video signals from the RF receiver and performs necessary operations or manipulations on the data. It can execute programmed instructions to extract relevant information, perform data filtering, or implement specific algorithms for further analysis.

Moreover, Arduino interprets the received video data and converts it into a format that can be understood and utilized by other components in the system. It can extract metadata, such as timestamp, camera information, or other relevant parameters embedded in the video stream. This interpretation allows for proper handling and further processing of the received video signals.

In addition, Arduino facilitates communication within the receiver section. It can interface with other components, such as the LCD display and piezo buzzer, to provide visual and audio feedback to the user.

IV.3.3. Video Processing

Video processing plays a crucial role in the security system, particularly in detecting and identifying potential threats or security incidents. In this context, the system utilizes advanced deep learning models, which is in our security system YOLO (You Only Look Once), for efficient and accurate video analysis. Moreover, to work with these models we have two parts:

- ✓ Simulation with the use of "Google Colaboratory" which we talked about it in detail in chapter three.
- ✓ The realization, which we are going to see it in this chapter.

We have used versions of yolov5, but before start using them we have to add the Yolo model to the Nvidia Jetson Nano (Figure IV.32).

Therefore, from files "Intelligent Monitoring System" we choose the "yolov5" file.

Figure IV. 32. Choosing Yolov5 File.

After that, we open the yolov5 file in terminal to work the model (Figure IV.33).

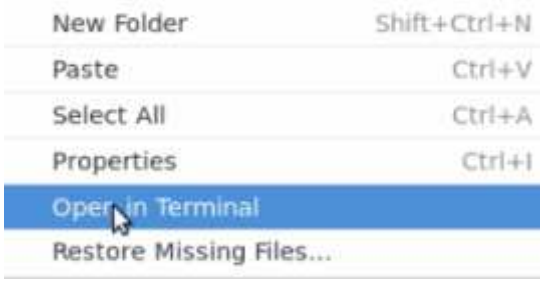


Figure IV. 33. Opening yolov5 in Terminal.

We add the file of weights, which is a file, contains almost all the data that the artificial intelligence algorithms will processing it from person's images and weapon images (Figure IV.34). Moreover, we choose this file especially because it is the best one experimentally.

Figure IV. 34. Weights File.

Since we have used deferent yolov5 models (Figure IV.35) in our project, we focused only on yolov5n models, which we use two applications:

Figure IV. 35. Yolov5 Models.

- ❖ In the first application, we use "yolov5n" model to detect persons.
- ❖ Moreover, the second application, we use "yolov5nWeapon" to detect weapons.

However, to detect persons or weapons, we go to the "yolov5 file" (Figure IV.32) choose "Open in Terminal" (Figure IV.33) then we get a window called "Nvidia Desktop" (Figure IV.36) which is the window where choose whenever we want to detect persons or weapons.

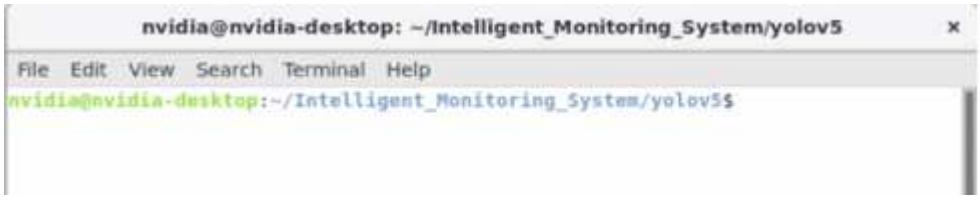


Figure IV. 36. Nvidia Desktop.

After that, we choose always from yolov5 file, the file "detect_Weapon_arduino" which is the file that we make changes on to specify the type of detect we want (Figure IV.37), and it is have the Arduino on its name because Nvidia Jetson Nano communicate serially with Arduino Uno (Figure IV.38).

Figure IV. 37. Detect Weapon Arduino File.

```
import serial
rd = serial.Serial("/dev/ttyACM0", baudrate=9600)
```

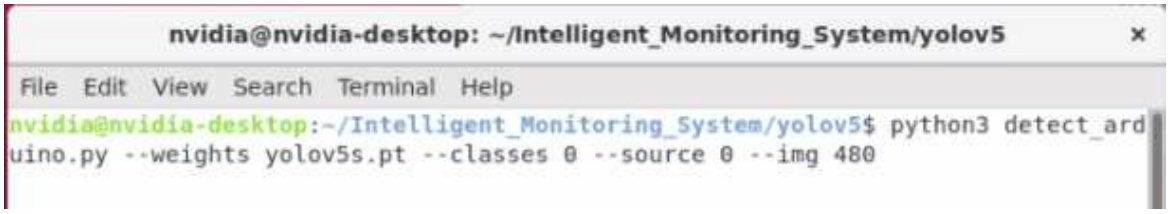
Figure IV. 38. Serial Communication between Nvidia Jetson Nan and Arduino Uno.

When we choose the file a new window opens (Figure IV.39.(a)), we copy the code and paste it in Nvidia Jetson Nano desktop then we start changing classes and sources as we want whenever person or weapon detection (Figure IV.39.(b)).

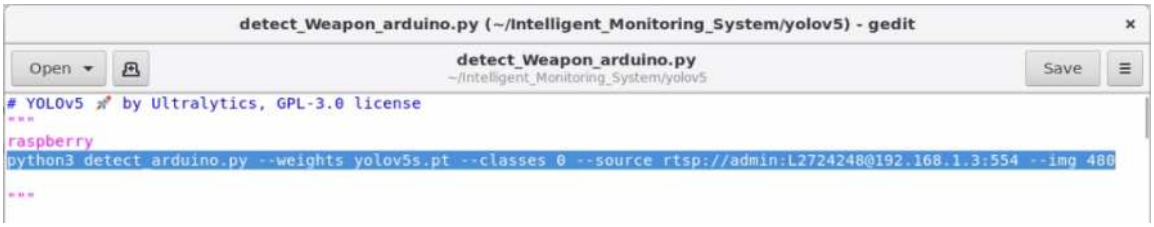
So to detect persons or weapons, we put:

- Sources are 0 or 1 (Depends on camera type).
- In addition, for the classes, if we do not want to specify a number we just delete them.

We simply click enter from keyboard PC then the camera interface will turn on to show us the monitored area (Figure IV.40).



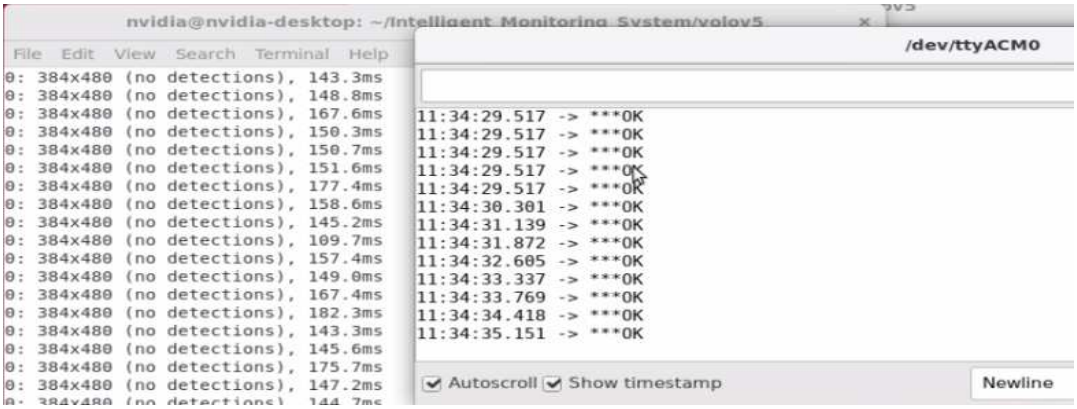
(a)



(b)

Figure IV. 39. (a) Detect Weapon Arduino file window, and (b) determine the type of detection.

(a)



(b)

Figure IV. 40. (a) Camera Interface, (b) Detection Interface.

The YOLO models used in the security system use deep learning algorithms and convolutional neural networks (CNNs) to efficiently process video images and make accurate predictions. These models have been trained on large datasets to detect and classify various objects, including people and weapons, with great accuracy.

By integrating YOLO models into the video processing pipeline, the security system can automate the detection process, significantly reducing reliance on manual surveillance. Real-time video analytics enable immediate threat detection, facilitating rapid response and mitigating potential security risks.

It is worth noting that YOLO models can be tuned and optimized based on specific security system requirements. This adjustment enables improved accuracy and performance, ensures reliable detection, and reduces false positives.

In general, the integration of YOLO models into the video processing stage of the security system improves its ability to detect people and possible weapons. This advanced level of analysis contributes to a more effective and proactive approach to security, enabling early detection of threats and ensuring the security of the monitored area.

IV.4. Project Prototype

In this section, we make assemblies of the previous assemblies that we have used to make the prototype of our project. In order to be as faithful as possible with the real system, we have used two parts for guarantee the exchange of information between the transmitter and the receiver using radio frequency communication.

For the first part (transmitter), we opted for the following components:

- Power supply,
- RF Transmitter nRF24l01,

- GSM900,
- Arduino Nano,
- Arduino Uno,
- Nvidia Jetson Nano B01
- An IP Camera.

For the second part (receiver), we opted for the following components:

- Power Supply,
- RF Receiver,
- Piezo Buzzer and
- LCD 16×2 display.

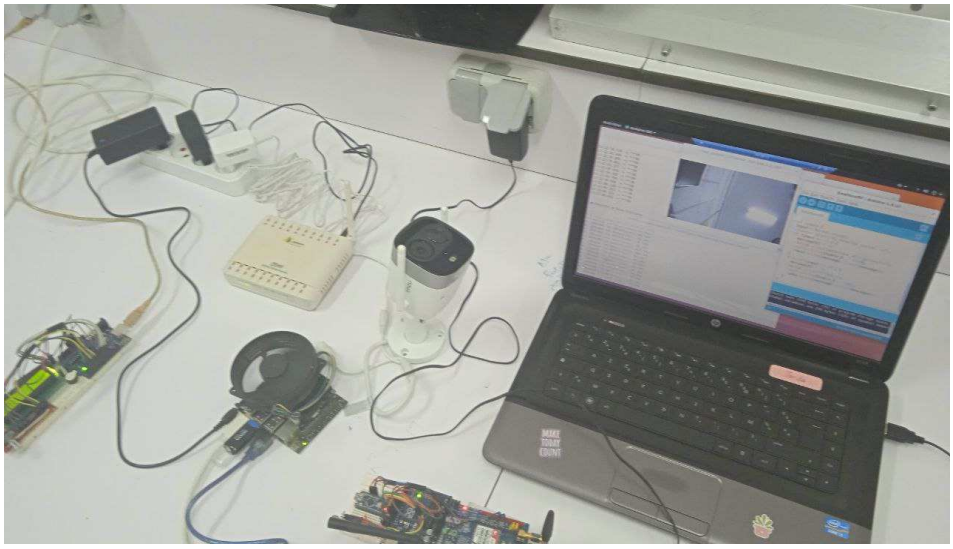


Figure IV. 41. Project Prototype.

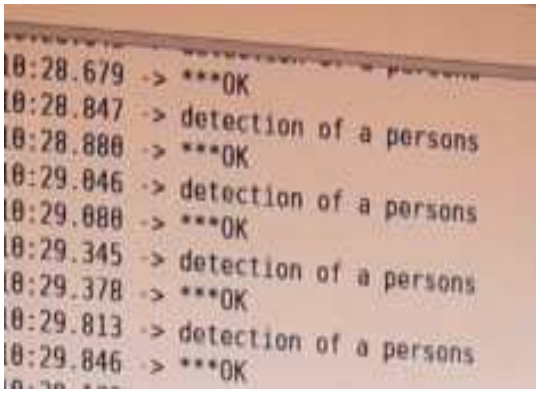
Firstly, we make assemblies of all the components (Figure IV.41) which is our project.

Secondly, we activate the two parts as mentioned in section (IV.3.1.3) then we start the part of integrate artificial intelligence methods and wireless transmission technologies to achieve real-time monitoring and security threat detection by video processing (Section IV.3.3). Therefore, we have two examples: person detection and weapon detection.

Finally, as depicted in Figure IV.49, we have the option to choose the detection type between persons or weapons. Based on our selection, we monitor the output of the IP Camera to determine whether it detects a person or a weapon.

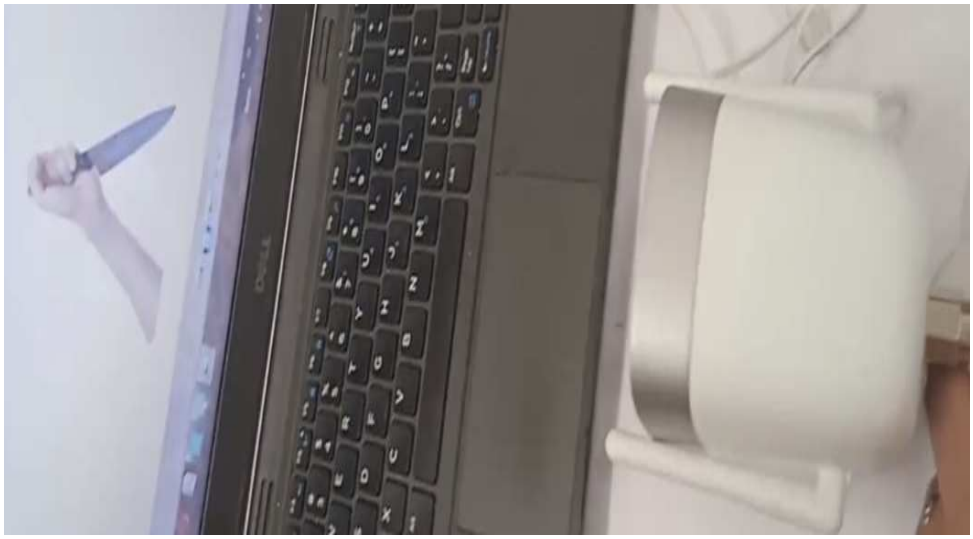
- If the IP Camera detects a person in the detection interface, we will see “person detection”.
- Moreover, it is the same for weapon detection; we will see “weapon detection” in the detection interface.

(a)

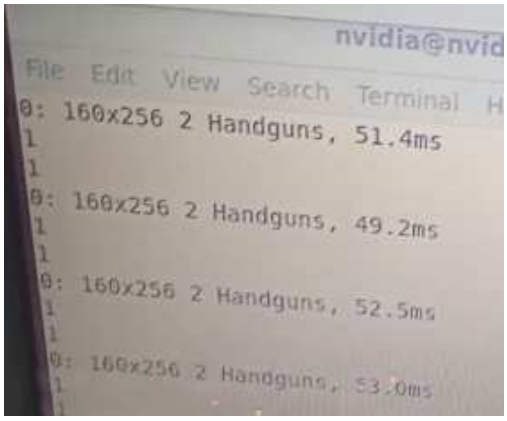


(b) (c)

Figure IV. 42. (a) Example of Person Detection, (b) and (c) Person Interface Detection.



(a)

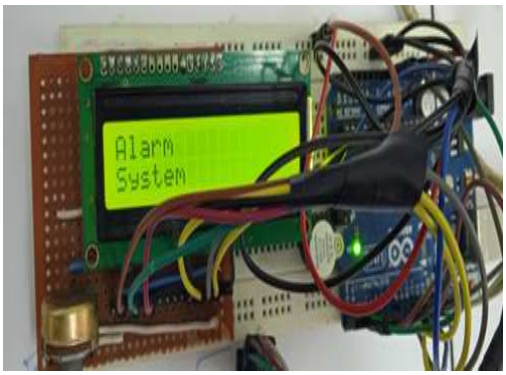


(b) (c)

Figure IV. 43. (a) Example of weapon detection, (b) and (c) Weapon Interface Detection.

Moreover, when the IP Camera detect person or weapon, the GSM module will send an alert (Figure IV.44), which is an SMS message or a call to the security center whether individuals or specific places such as banks. In the same time, the status of the system will be displayed using the LCD display as shown in below (Figure IV.45).

Figure IV. 44. GSM Alert.



(a) (b)

Figure IV. 45. (a) Safety System, (b) Alert System.

The alarm system process involves two levels of detection:

- 0 – Safety system (No Detection)
- 1 – Alert system (Warning)

On the first level, no alarm will be triggered in both a call and buzzer. In the second level of detection, a call alarm will be sent to the user and at the same time, the buzzer will be triggered when a weapon or a person detected.

IV.5. Conclusion

In this last chapter, we have provided a comprehensive overview of our security system prototype, discussing the working environment and the electronic components used. We also presented the efficiency and performance of our algorithm for person and weapon detection, show casing its accuracy and reliability. Furthermore, we conducted several tests to gain deeper insights into the functionality of our project.

The successful integration of wireless communication, Python programming, and Nvidia Jetson Nano algorithms has been instrumental in creating an advanced security system. Wireless communication has played a pivotal role in establishing seamless connectivity between system components, facilitating real-time transmission of video data for prompt analysis and response. This utilization of wireless transmission technologies has provided flexibility and scalability, enabling easy expansion and integration with other devices and networks.

Python programming has served as a versatile platform for implementing complex algorithms, particularly in video processing. The integration of YOLOv5 models, specifically YOLOv5n, has significantly enhanced the system's ability to accurately detect and classify persons and weapons, greatly improving overall security outcomes.

At the core of the processing unit, the Nvidia Jetson Nano has demonstrated its impressive power and efficiency. With its high computational capabilities and dedicated GPU, it has enabled real-time analysis and handling of deep learning models. This has resulted in swift and accurate security alerts and actions, further augmenting the system's overall performance.

The combined achievements of wireless communication, Python programming, and Nvidia Jetson Nano algorithms have yielded highly satisfying results. The system's precise person and weapon detection, along with its real-time capabilities, have exceeded expectations, reinforcing security and threat mitigation efforts. Furthermore, the successful integration of these technologies paves the way for future advancements, such as incorporating facial

recognition, behavior analysis, or crowd monitoring, to effectively address evolving security challenges.

IV.6. References cited in Chapter IV

[1] P. Kang and A. Somtham, "An Evaluation of Modern Accelerator-Based Edge Devices for Object Detection Applications," *Mathematics,* vol. 10, p. 4299, 2022.

[2] A. Kurniawan, "IoT Projects with NVIDIA Jetson Nano," *IoT Projects with NVIDIA Jetson Nano,* 2021.

[3] F. N. UZUN, M. KAYRICI, and B. AKKUZU, "Nvidia jetson nano development kit," *Programmable Smart Microcontroller Cards,* p. 82, 2021.

[4] P. Dai, "GSM REMOTE CONTROL HEATER," Degree Program in Information Technology, VAASAN AMMATTIKORKEAKOULU UNIVERSITY OF APPLIED SCIENCES, 2014.

General Conclusion

General Conclusion

Security is always a main concern in every domain. Alarm detection provides important applications for computer vision in solving a variety of problems and challenges facing our society.

The objective of this work was to combine between wireless transmission technologies and artificial intelligence approaches and to implement the prototype for the development of smart monitoring security system specifically detecting person and weapons.

The theoretical part of this work presented an overview of existing security systems and discussed two different wireless communication technologies: radio frequency communication and the GSM system. In addition, the fundamental concepts of various machine learning, and deep learning methods that utilized in our project.

There are many detection methods in existence but the accuracy, rapidity, and efficiency of detection are not good enough. The YOLO architecture has demonstrated promising results in various object detection tasks offering both high speed and good performance. In third chapter after we present the software aspects including tools and libraries we used in our system, we evaluated five versions of the YOLOv5 model namely YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5n andYOLOv5x. After the simulation and comparison, the YOLOv5n model was chosen as the best option for the detection task.

In the second phase, we tested the robustness and performance of the artificial intelligence methods developed in the simulation phase to be used in real-world scenarios. To achieve this, we created a prototype of our monitoring and security system, incorporating the tested AI algorithms.

Let's discuss the components used and their functions within the system. Our system is divided into two main sections: the transmitter section and the receiver section.

The transmitter section comprises several key components: the RF Transmitter, GSM900 module, Arduino Nano, Arduino Uno, Nvidia Jetson Nano B01, and an IP Camera. Each component serves a specific purpose in the overall functionality of the transmitter section. The RF Transmitter facilitates wireless communication, allowing data transmission from the system to the receiver. The GSM900 module enables cellular connectivity, enabling the system to send notifications and alerts. The Arduino Nano and Arduino Uno provide processing power and control for the system, facilitating data processing and decision-making. The Nvidia Jetson Nano B01 enhances computational capabilities, enabling complex AI algorithms to run efficiently. Finally, the IP Camera captures real-time video footage, which is an essential input for the AI algorithms to detect security threats accurately.

Moving on to the receiver section, it consists of the RF Receiver, Piezo Buzzer, and LCD display. The RF Receiver receives the wireless signals transmitted by the transmitter section, allowing for seamless communication between the two sections. The Piezo Buzzer acts as an audible alert mechanism. The LCD display provides a visual interface for monitoring and displaying relevant information and system status updates.

The combination allowed us to build effective monitoring and security system capable of quick detection and response to security threats:

- Implementing the YOLOv5n model in the Nvidia Jetson Nano offered significant advantages in terms of real-time object detection.
- At the management level, our system reduce costs, control resource allocation, aid decision-making, and even provide opportunities for early intervention to mitigate insider threats.
- This system can be employed in institutions such as banks and it can be employed in police stations and prisons in order to enhance the security of the criminals.

When we look at the prospects for our project, several key aspects come to light. First, the affordability of the system is a crucial consideration. We have strived to create a cost-effective solution without compromising on performance and accuracy. By using readily available components and open source technologies, we want to make our security system accessible to a wide range of users.

In addition, the real-time capability of our system is a major advantage.Through the ability to monitor and respond to security threats in real time, we provide users with timely information and the ability to take quick action.This contributes to the overall security of individuals and organizations.

In summary, realizing a security system using artificial intelligence methods and wireless transmission technologies has been an exciting and rewarding journey. The realization phase demonstrated the power of real-time monitoring, with YOLOv5n proving to be an excellent tool. Practical implementation has shown the perfect integration of hardware components and the effectiveness of our system in object detection and wireless communication.

For the future, there are numerous prospects for further development and expansion. One research approach is the integration of cloud services and the connection of the system to the Internet. This would allow remote access and control and improve the flexibility and scalability of the system. Additionally, the incorporation of advanced machine learning techniques such as facial recognition and behavioral analysis can enable deeper insights and more sophisticated threat detection.