الجمهورية الشعبية الديمقراطية الجزائرية
**People's Democratic Republic of Algeria**
وزارة التعليم العالي و البحث العلمي
**Ministry of Higher Education and Scientific Research**
**جامعة الدكتور مولاي الطاهر سعيدة**
**University Of Dr.Moulay Taher**
**Saida**

# Master's Thesis

To obtain the diploma of **Master's Degree**

Field of Study: **Electrotechnic department**

Specialization: **Automatic And System**

## Theme

---

# Real-time Management In Traffic Car Parking

---

Presented by
**Hafsi Nadir**
**Drif Hichem**

Defended on: **Juin, 2025**
*In front of the jury composed of*

| | |
|---|---|
| **Dr. MEKKAOUI Mohamed** | President of the Jury |
| **Pr. Mostefai Mohamed** | Thesis Supervisor |
| **Dr. MOSTEFAI Lotfi** | Examiner |

*Academic Year: 2024/2025*

# *DEDICATION*

*With love and appreciation, I dedicate this to my family, who have always supported and encouraged me. I appreciate your unwavering faith in me. To my friends for their sincere*

*friendship and comforting presence. Your support has been invaluable throughout this journey. To my teachers and mentors for their guidance and invaluable advice. Your wisdom and*

*expertise have been an immeasurable source of inspiration. To all those who contributed*

*directly or indirectly to the completion of this work. Your help and support have been essential. Thank you all.*

**HAFSI NADIR**

# *DEDICATION*

*With love and appreciation, I dedicate this to my family, who have always supported and encouraged me. I appreciate your unwavering faith in me. To my friends for their sincere*

*friendship and comforting presence. Your support has been invaluable throughout this journey. To my teachers and mentors for their guidance and invaluable advice. Your wisdom and*

*expertise have been an immeasurable source of inspiration. To all those who contributed*

*directly or indirectly to the completion of this work. Your help and support have been essential. Thank you all.*

**DRIF HICHEM**

# ACKNOWLEDGMENTS

Completing this thesis would not have been possible without the help and encouragement of so many people. To each of you, whether you played a major role or offered quiet support along the way, thank you.

My deepest thanks go to my advisor, Mr. MOSTFAI Mohammed, for his trust, patience, and invaluable guidance. His expertise and dedication not only shaped this research, but also inspired me to push my limits. I am truly grateful for his unwavering support.

I am also very grateful to my thesis committee members for taking the time to review my work and share their thoughtful feedback. Your insights have been incredibly valuable.

To my fellow students and friends in automation, thank you for the lively discussions, the moral support, and for making this journey much more enjoyable. Your camaraderie kept me motivated even during the toughest moments.

Most of all, I owe endless gratitude to my family. Your constant encouragement, patience, and belief in me carried me through every challenge. Without you, none of this would have been possible.

From the bottom of my heart I thank you all.

# ABSTRACT

At urban intersections or access control points (such as automatic barriers), traffic fluidity and safety generally require the use of physical position or presence sensors. However, these sensors are expensive, rigid, and limited in number, which complicates the implementation of effective management in budget-restricted areas.

In this work, we propose an intelligent alternative using a camera mounted above intersections or portals to monitor traffic in real-time. The captured images are processed by a computer vision algorithm based on YOLOv12 (You Only Look Once), a high-performance object detection model. This system enables precise real-time identification and counting of vehicles in the scene.

Using the collected data (vehicle count and type), the system can :

Dynamically adjust traffic light cycles according to traffic density

Automate barrier opening/closing operations

Detect license plates (via OCR) for access logging and permission verification

This project provides an economical, intelligent, and scalable solution to modernize traffic management and access control systems without requiring expensive sensor infrastructure.

**Keywords :**
Traffic lights, Automatic barrier, Vehicle detection,
Computer vision, YOLOv12, Microcontroller

# Résumé

Dans les carrefours urbains ou les points de contrôle d'accès (comme les barrières automatiques), la uidité et la sécurité du tra c nécessitent généralement l'utilisation de capteurs physiques de position ou de présence. Cependant, ces capteurs sont coûteux, rigides et limités en nombre, ce qui complique la mise en œuvre d'une gestion ef cace dans des zones à budget restreint.

Dans ce travail, nous proposons une alternative intelligente à ces capteurs en utilisant une caméra placée au-dessus des intersections ou des portails pour surveiller le tra c en temps réel. Les images capturées sont traitées par un algorithme de vision par ordinateur basé sur YOLOv12 (You Only Look Once), un modèle de détection d'objets à haute performance. Ce dernier permet d'identi er et de compter avec précision les véhicules présents dans la scène, en temps réel.

Grâce aux données recueillies (nombre et type de véhicules), le système est capable :

- d'ajuster dynamiquement les cycles des feux tricolores selon la densité du tra c,
- d'automatiser l'ouverture et la fermeture des barrières,
- de détecter les plaques d'immatriculation (reconnaissance de texte OCR) pour journaliser les accès ou véri er les autorisations d'entrée.

Ainsi, ce projet offre une solution économique, intelligente et évolutive pour moderniser les systèmes de gestion du tra c et de contrôle d'accès, sans dépendre d'une infrastructure de capteurs coûteuse.

Mots-clés :
Feux tricolores, Barrière automatique, Détection de véhicules, Reconnaissance d'images, YOLOv12, Microcontrôleur

# LIST OF ABBREVIATIONS

| Abbreviation | English | French |
|---|---|---|
| APS | Automated Parking System | Système de Parking Automatisé |
| IoT | Internet of Things | Internet des Objets |
| NPR/LPR/RPR | License Plate Recognition | Reconnaissance de Plaques d'Immatriculation |
| AI | Arti cial Intelligence | Intelligence Arti cielle |
| NFC | Near Field Communication | Communication en Champ Proche |
| GPS | Global Positioning System | Système de Positionnement Global |
| CV | Computer Vision | Vision par Ordinateur |
| CNN | Convolutional Neural Network | Réseau de Neurones Convolutifs |
| OCR | Optical Character Recognition | Reconnaissance Optique de Caractères |
| GAN | Generative Adversarial Network | Réseaux Antagonistes Génératifs |
| SIFT | Scale-Invariant Feature Transform | Transformée des Caractéristiques Invariantes à l'Échelle |
| SURF | Speeded-Up Robust Features | Caractéristiques Robustes Accélérées |
| ORB | Oriented FAST and Rotated BRIEF | BRIEF Orienté et Tourné |
| ML | Machine Learning | Apprentissage Automatique |
| DL | Deep Learning | Apprentissage Profond |
| SRCNN | Super-Resolution Convolutional Neural Network | Réseau de Neurones Convolutifs à Super-Résolution |
| ESRGAN | Enhanced Super-Resolution Generative Adversarial Network | Réseaux Antagonistes Génératifs à Super-Résolution Améliorée |
| LIDAR | Light Detection and Ranging | Détection et Télémétrie par Lumière |
| SVM | Support Vector Machine | Machine à Vecteurs de Support |
| YOLO | You Only Look Once | Vous Ne Regardez Qu'Une Fois |
| R-ELAN | Residual Ef cient Layer Aggregation Networks | Réseaux d'Agrégation de Couches Ef caces Résiduels |
| COCO | Common Objects in Context | Common Objects in Context |
| mAP | Mean Average Precision | Précision Moyenne |

Continued on next page

| Abbreviation | English | French |
|---|---|---|
| FLOPs | Floating Point Operations | Opérations en Virgule Flottante |
| OBD | Oriented Object Detection | Détection d'Objets Orientés |
| ESP8266 | ESP8266 Microcontroller | Microcontrôleur ESP8266 |
| LED | Light Emitting Diode | Diode Électroluminescente |
| RTOS | Real-Time Operating System | Système d'Exploitation Temps Réel |
| USB | Universal Serial Bus | Bus Série Universel |
| UART | Universal Asynchronous Receiver-Transmitter | Récepteur-Émetteur Asynchrone Universel |
| WCET | Worst-Case Execution Time | Temps d'Exécution Pire Cas |
| ROS | Robot Operating System | Système d'Exploitation Robotique |
| EEG | Electroencephalogram | Électroencéphalogramme |
| QC | Quantum Computing | Informatique Quantique |

# TABLE OF CONTENTS

TABLE OF CONTENTS

DEDICATION i

ACKNOWLEDGMENTS iii

ABSTRACT iv

LIST OF ABBREVIATIONS vii

LIST OF FIGURES xvi

LIST OF TABLES xviii

General Introduction 1

1 Intelligent Parking : Concepts and Typologies 2
1.1 Introduction . . . . . 3
1.2 De nition . . . . . 3
1.3 The Different Types Of Smart Parking : . . . . . 5
1.3.1 Parking with sensors on the ground . . . . . 5
1.3.2 Parking based on computer vision . . . . . 6
1.3.3 Smart phone application based smart parking system . . . . . 8
1.3.4 Automated parking system . . . . . 9
1.3.5 Smart City integrated car parks . . . . . 11
1.4 Advantages of smart parking : . . . . . 12
1.5 Disadvantages and limitations . . . . . 13
1.6 Conclusion . . . . . 13

2 Computer Vision and Its Applications 15
2.1 INTRODUCTION . . . . . 16
2.1.1 De nition of computer vision . . . . . 16
2.2 Computer Vision Objectives . . . . . 17
2.3 Technological Foundations . . . . . 25
2.3.1 Image Processing . . . . . 25
2.3.2 Machine and Deep Learning . . . . . 28
2.3.3 Sensors and Hardware . . . . . 29
2.4 Methods and Algorithms . . . . . 29
2.4.1 Image Classi cation . . . . . 29
2.4.2 Object Detection . . . . . 31
2.4.3 Semantic and Instance Segmentation . . . . . 31

2.4.4 Object Tracking . . . . . 31
2.4.5 Generation and Reconstruction . . . . . 32
2.5 Applications of Computer Vision . . . . . 33
2.5.1 Manufacturing . . . . . 33
2.5.2 Healthcare . . . . . 33
2.5.3 Transportation . . . . . 33
2.5.4 Trade . . . . . 33
2.5.5 Security and Surveillance . . . . . 34
2.5.6 Agriculture . . . . . 34
2.6 Challenges and Perspectives : . . . . . 34
2.7 Future Prospects : . . . . . 35
2.8 Conclusion . . . . . 36

3 Application of Computer Vision to Access Management at the University of Saida Parking 37
3.1 Introduction . . . . . 38
3.2 Speci cations . . . . . 38
3.3 Application of YOLO . . . . . 40
3.3.1 Functional diagram . . . . . 42
3.3.2 Diagram with recognition of plates . . . . . 43
3.3.3 Dataset preparation : . . . . . 44
3.4 Trials . . . . . 48
3.5 Limits and prospects . . . . . 49
3.5.1 Limits : . . . . . 49
3.5.2 Prospects : . . . . . 49
3.6 Conclusion . . . . . 50

4 Development of the university access control prototype using computer vision 51
4.1 Presentation of the prototype . . . . . 52
4.1.1 Flowchart . . . . . 53
4.1.2 Simpli ed electrical diagram . . . . . 54
4.2 Technical details of the codes . . . . . 60
4.2.1 ESP8266 Code . . . . . 60
4.2.2 Code Python . . . . . 62
4.3 Preparation of the datasets . . . . . 64
4.4 Learning . . . . . 68
4.5 Test and trial . . . . . 71

4.5.1 Laboratory tests : . . . . . 71
4.5.2 Practical tests : . . . . . 71
4.6 Limits and Defects . . . . . 82
4.6.1 Speci c Defects Observed in Trials . . . . . 83
4.7 Conclusion . . . . . 84

General Conclusion 85

Bibliography 88

Appendix A 93

Appendix B 108

# LIST OF FIGURES

1.1 Navigating the urban parking landscape . . . . . 3
1.2 Parking garage Entry/Exit. . . . . 4
1.3 Parking entry. . . . . 4
1.4 Smart parking environment. . . . . 5
1.5 Ground sensor. . . . . 6
1.6 Type of ground sensor. . . . . 6
1.7 Parking lot with AI analysis. . . . . 7
1.8 Real-time parking space monitoring. . . . . 7
1.9 Digital parking experience. . . . . 8
1.10 Mobile app for parking lot enable. . . . . 8
1.11 Smart parking tower. . . . . 9
1.12 Automated parking system. . . . . 10
1.13 Parking by machine. . . . . 10
1.14 Smart cities's parking. . . . . 11
1.15 Data ow in urban spaces. . . . . 11

2.1 Human Vision vs. Computer Vision [1]. . . . . 18
2.2 Using object recognition to identify different categories of objects. . . . . 19
2.3 Optical illusions regarding the problem of image segmentation . . . . . 20
2.4 Human segmentation of example images ; source : Svetlana Lazebnik . . . . . 20
2.5 The superpixels allow faster computations by clustering pixels [2]. . . . . 21
2.6 Object tracking camera [3]. . . . . 23
2.7 3D reconstruction . . . . . 24
2.8 3D reconstruction . . . . . 24
2.9 Example of High-pass Image ltering [4] . . . . . 26
2.10 Example of Low-pass Image ltering [4] . . . . . 26
2.11 Sample problems of bad edge detectors . . . . . 27
2.12 Certain areas of the brain react to edges ; the line drawings are as recognizable as the original image [5]. . . . . 28
2.13 Instance Segmentation by Clustering Technique . . . . . 31
2.14 Comparisons of bicubic-upsampled, ESRGAN, RealSR, and Real-ESRGAN results on real-life images . . . . . 32

3.1 Access of the university of Saida . . . . . 39
3.2 Access's barriers of the university. . . . . 39

3.3 Comparison of heat maps between YOLOv10, YOLOv11,and the proposed YOLOv12. Compared to the advance YOLOv10 and YOLOv11, YOLOv12 demonstrate a clearer perception of objects in the image. All the results are obtained using the X scale models. Zoom into compare the details. . . . . 41
3.4 Detection performance. (COCO val2017) . . . . . 42
3.5 Functional diagram of the project. . . . . 42
3.6 The steps of an RPR system. . . . . 43
3.7 Neural network-based parking system real-time license plate detection . . . . . 43
3.8 Diagram with recognition of plates. . . . . 44
3.9 dataset preparation. . . . . 45
3.10 Dataset label Distribution. . . . . 45
3.11 Con dence curve. . . . . 46
3.12 Annotation (Labelling with YOLO) . . . . . 46
3.13 Application of transformations. . . . . 47
3.14 Division of dataset result. . . . . 47
3.15 Storage of result. . . . . 47
3.16 Trial in real time (Personal view). . . . . 48
3.17 Trial in real time (System view). . . . . 48

4.1 Photo of the prototype showing the miniature road, the barrier, the traf c light, the camera, and the miniature cars. . . . . 52
4.2 Prototype operation diagram . . . . . 53
4.3 CAM (Elecom Ucam-dln130tpn) . . . . . 55
4.4 ESP8266 Card (NodeMCU) . . . . . 56
4.5 Pinout of the NodeMCU ESP8266 . . . . . 56
4.6 View of a servo motor . . . . . 58
4.7 LED . . . . . 59
4.8 Bread Board . . . . . 59
4.9 Simpli ed electrical diagram of the prototype . . . . . 60
4.10 ESP8266 Code operation diagram . . . . . 61
4.11 Python Code diagram . . . . . 62
4.12 Python Code diagram (License Plate Detection) . . . . . 63
4.13 Data collection . . . . . 64
4.14 Annotation (Labelling) . . . . . 65
4.15 Dataset Label Distribution . . . . . 65
4.16 Dataset Label Correlogram/Pair Plot . . . . . 66
4.17 Cleaning methode . . . . . 67

4.18 Augmentation Application . . . . . 68
4.19 The training results . . . . . 69
4.20 Results training with Yolov12 . . . . . 69
4.21 File of results training with Yolov12 illustrate the number of epoch used, Time, Box loss, Precision. . .. . .ect. . . . . 70
4.22 Precision-con dence Curve. . . . . 70
4.23 overhead view of a prototype smart road and traf c management system . . . . 72
4.24 First trial in real time (Personal view) . . . . . 73
4.25 First trial in real time (system view) . . . . . 74
4.26 Second trial (Personal view) . . . . . 76
4.27 Second trial of our project . . . . . 77
4.28 Final trial . . . . . 79

# LIST OF TABLES

4.1 Components of prototype . . . . . 54

# General Introduction

In an era de ned by digital transformation and the rapid advancement of arti cial intelligence, access control and traf c management systems are experiencing a signi cant evolution. Traditional solutions that depend on physical sensors or magnetic cards present numerous limitations particularly in terms of cost,  exibility, and maintenance. These challenges are especially pronounced in environments such as universities or low-resource areas, where the demand for secure and ef cient traf c  ow is continually increasing.

The growing volume of vehicular traf c, combined with accelerated urbanization, has raised serious concerns regarding congestion, inef ciency, and safety across both public and private transportation infrastructures. In many settings such as university campuses, parking lots, or small urban zones traditional traf c management systems rely heavily on  xed-timer traf c lights and expensive physical sensors (e.g., inductive loops or infrared detectors). These conventional methods often lack the ability to adapt to real-time traf c conditions, resulting in unnecessary delays, driver frustration, increased fuel consumption, and environmental harm.

With limited budgets and aging infrastructure, many institutions are unable to adopt dynamic traf c regulation systems. This underscores the urgent need for smarter and more cost-effective alternatives. Recent developments in arti cial intelligence especially in machine learning, computer vision, and edge computing—have paved the way for intelligent transportation systems that are both adaptive and scalable.

This project proposes an AI-powered, camera-based access control system that replaces traditional physical sensors with real-time vehicle detection using computer vision techniques (speci cally, YOLO). Leveraging affordable components such as a camera and ESP8266, along with Wi-Fi communication and automated control of gates and traf c signals, our system offers a dynamic and ef cient access management solution particularly well-suited for university campuses.

This thesis is organized into four chapters :

Chapter 1 presents an overview of smart parking systems and introduces key concepts in traf c management, along with a review of related work.

Chapter 2 focuses on intelligent vision systems, providing background on our proposed solution and the underlying technologies.

Chapters 3 and 4detail our proposed system, including the practical implementation and a description of the core components used to build a model for parking access control.

Finally, the thesis concludes with a summary of  ndings and perspectives for future development.

CHAPTER

1

# Intelligent Parking : Concepts and Typologies

## 1.1 Introduction

Smart parking uses technology to optimize the availability of spaces and simplify the parking process. Its origins date back to the early 2010s.

Thanks to intelligent systems, smart parking makes it easier to locate vacant spaces. It improves the experience of drivers and the  uidity within the parking lot. This innovative solution enhances the ef ciency of parking management and is perfectly adapted to the needs of modern cities.

## 1.2 De nition

Smart parking revolutionises the search for parking spaces thanks to intelligent management supported by modern technologies. It optimizes the parking experience by making it easier to locate vacant spaces, improving the  uidity and management of parking.[6] It can also be de ned as an IoT-powered system that relies on sensors and/or cameras to gather data regarding the availability of spots in a particular parking area. To interact with such a system, a driver needs the right mobile application (in some cases such applications may also have a web version). As a rule, such apps allow users to book a place in advance and pay to park there.[7]

FIGURE 1.1 – Navigating the urban parking landscape .

FIGURE 1.2 – Parking garage Entry/Exit.

FIGURE 1.3 – Parking entry.

FIGURE 1.4 – Smart parking environment.

## 1.3 The Different Types Of Smart Parking :

Intelligent parking system uses sensors or other technologies to determine the availability of city parking .These include the following :

### 1.3.1 Parking with sensors on the ground

Parking sensors are inserted underground and transmit data to devices operated by parking inspectors, indicating when vehicles have been parked in an area for longer than signed. It is important that car parks in busy shopping centres and near strip shops and businesses are vacated regularly to give everyone a chance of getting a car park and to ensure a steady stream of customers for local businesses. In-ground sensors have largely been well received by the likes of traders, which ensures a fair turnover for businesses and a fair go for users of the activity centres. Sensors also enable Council to provide reliable enforcement and monitor parking trends.[8]

FIGURE 1.5 – Ground sensor.

FIGURE 1.6 – Type of ground sensor.

### 1.3.2 Parking based on computer vision

Using computer vision for parking, also known as smart parking, can revolutionize urban mobility by optimizing the utilisation of a given space. Of the many upshots, this can reduce traf c congestion and enhance overall parking ef ciency.

Through real-time monitoring and analysis of parking areas, computer vision technology can streamline the parking experience for the driver, providing them with accurate information on available spaces, minimizing search times, and contributing to a more sustainable and user-friendly urban environment.

For businesses or city planners, computer vision can help them identify blindspots in their infrastructure. Maybe certain areas of congestion, or parking bays with minimal use, or a particular area prone to longer wait times, for example.

With the aid of IoT sensors and CCTV footage for video analytics, the presence or absence of vehicles can easily be detected without the need for manual checks. This information can then be transmitted and analyzed, automatically identifying occupancy insights to detail, to use one example, the number of free parking spaces in a given area.[9]

FIGURE 1.7 – Parking lot with AI analysis.

FIGURE 1.8 – Real-time parking space monitoring.

### 1.3.3 Smart phone application based smart parking system

A signi cant percentage of smart parking systems deploy Android or IOS applications so that end users can get information about smart parking facilities. Smartphone applications also provide a graphical interface for the user to interact with the system. In addition, to receive real-time parking status information, obtain a vehicle guide to the nearest parking area or parking lot, reserve parking, Collect parking information and pay for parking fees through online services or Near Field Communication smart phone technology. [10]

FIGURE 1.9 – Digital parking experience.

FIGURE 1.10 – Mobile app for parking lot enable.

### 1.3.4 Automated parking system

An automated (car) parking system (APS) is a mechanical system designed to minimize the area and/or volume required for parking cars. Like a multi-story parking garage, an APS provides parking for cars on multiple levels stacked vertically to maximize the number of parking spaces while minimizing land usage. The APS, however, utilizes a mechanical system to transport cars to and from parking spaces (rather than the driver) in order to eliminate much of the space wasted in a multi-story parking garage. While a multi-story parking garage is similar to multiple parking lots stacked vertically, an APS is more similar to an automated storage and retrieval system for cars. Parking systems are generally powered by electric motors or hydraulic pumps that move vehicles into a storage position.

APS are also generically known by a variety of other names, including : automated parking facility (APF), automated vehicle storage and retrieval system (AVSRS), car parking system, mechanical parking, and robotic parking garage [11].

FIGURE 1.11 – Smart parking tower.

FIGURE 1.12 – Automated parking system.

FIGURE 1.13 – Parking by machine.

### 1.3.5 Smart City integrated car parks

The concept of smart parking is derived from the Smart City (or smart city), applied to parking management. It is about using digital technologies to optimize vehicle parking, and therefore urban services in an indirect way.

In concrete terms, intelligent car parks use the data generated by new technologies to determine the number of available parking spaces, the exceeding of the occupancy limit and the registration plates of the vehicles present.

To collect these data sets, car parks are equipped with sensors connected to software processing actionable information. Various intelligent technologies are then used, such as optical counting to know the number of places available, vehicle guidance to allow drivers to quickly navigate to empty parking spaces or license plate reading to identify vehicle[12].

FIGURE 1.14 – Smart cities's parking.

FIGURE 1.15 – Data  ow in urban spaces.

## 1.4 Advantages of smart parking :

Smart parking allows for “bulking”, that is to say playing with different parking needs, depending on the type of users. In practice, the majority of car parks are never 100% full all the time. To popularize, for example, residents of the neighborhood who do not work nearby will take their car and will not be parked in the smart parking the day. On the contrary, those who come to work in the neighborhood will come to park during the day and will not be there, for example on weekends. This is where connected car parks also have a role to play : they allow us to build smaller spaces but that will bene t just as many people, all thanks to the data collected through technology.

Smart parking simpli es the process of  nding a parking space. Drivers can easily locate vacancies through real-time applications or billboards.

By guiding drivers to available parking spaces, smart parking solutions reduce traf c congestion and congestion inside the car park. This unnecessary traf c reduction improves traf c  ow. It limits the time spent looking for a place and promotes intelligent mobility, more ef cient mobility for everyone around.

Smart parking spaces can be equipped with advanced security features to ensure peace of mind for users. This is the case, for example, of the license plate reading system to track vehicles. Smart parking also plays a role in reducing carbon emissions and other pollutants. By minimizing the time spent searching for a parking space, smart parking contributes to a cleaner and more sustainable environment, limiting unnecessary travel.

The data collected allows for better management and more ef cient allocation of available parking space. This translates into an increase in parking capacity. This advantage is particularly valuable in areas of high demand, such as shopping malls.

Thanks to the data collected by intelligent parking systems, it is possible to implement more ef cient pricing strategies. Companies can introduce variable rates depending on peak hours and parking times. This allows them to maximize the revenue generated by parking.[6]

## 1.5 Disadvantages and limitations

One of the main disadvantages of smart parking is the high cost of its implementation. The installation of sensors, cameras and communication infrastructure can be a signi cant  nancial investment. Small and medium-sized enterprises sometimes  nd it dif cult to justify these costs, especially if they have budget constraints. Costs that can nevertheless be amortized because thanks to the concept of crowding, smaller car parks are built.

Intelligent parking systems require ongoing maintenance to ensure they are functioning properly. Electronic components may require repairs and software updates. Maintenance managemen generates additional operational costs.

The implementation of a smart parking solution entails an adaptation period. System management staff and users will need to familiarize themselves with the new interfaces. This initiation stage entails a gradual learning phase and the need to make adjustments.

Intelligent parking systems collect and process sensitive data. This includes vehicle information and parking habits. Companies must therefore ensure that the con dentiality of this data is maintained and comply with data protection regulations.[6]

## 1.6 Conclusion

While sensor-based smart parking systems have been widely used, they come with notable drawbacks such as high installation costs, limited functionality, and maintenance challenges. On the other hand, computer vision-based systems offer a more scalable,  exible, and data-rich alternative. They not only detect parking occupancy but also provide valuable insights like vehicle type, license plate recognition, and rule enforcement—all with fewer physical components.

As cities and businesses move toward smarter infrastructure, computer vision stands out as a more ef cient and future-ready solution for smart parking.

The second chapter will delve into how machine vision is applied to the management of intelligent parts. It will offer an in-depth analysis of the different techniques and technologies

used to incorporate machine vision into the tracking, handling, and optimization of intelligent parts within production or maintenance work ows, highlighting their effects on improving ef ciency, accuracy, and automation in processes.

CHAPTER

2

# Computer Vision and Its Applications

## 2.1 INTRODUCTION

Computer vision, also known as machine vision, is a subdiscipline of arti cial intelligence (AI) that enables machines to analyze and interpret images or videos in ways comparable to human visual perception. It leverages computer science, image processing, and machine learning and deep learning techniques to extract relevant information from visual data. This chapter examines the basic principles of computer vision, its major technologies, and its various applications in industrial and social sectors.

### 2.1.1 De nition of computer vision

Computer vision, also abbreviated as CV, can be de ned as the  eld of study that aims to develop techniques to help computers “see” and understand the content of digital images such as photographs and videos. The concept has been around for more than 50 years but has only recently begun to see a resurgence of interest in how machines `see' and how it can be used to build products for businesses and consumers.

Computer Vision is a  eld of Arti cial Intelligence and Computer Science. By giving computers the ability to gain a visual understanding of the world, we are able to apply the technology to aid us in several real world applications such as autonomous vehicles, facial recognition, Google Lens, etc.

Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, and image restoration, some of which will be covered in this article.

Machine Learning is largely used to develop computer vision, so if you haven't checked out our article on machine learning , we recommend you do so to have some background information for this article.

The internet is comprised of text and images. Indexing and searching text is relatively straightforward but when it comes to images, computers need to know what the images contain. Up until recently, the content of images and video were best described using the meta descriptions

provided by the user who uploaded the media. In order to expand on the amount of information we can extract from images, we need computers to `see' an image and understand the content. For humans this is a relatively easy task, but for computers, it's an entirely new challenge that would take decades to develop from the moment the idea were suggested in the 50s. [13]

## 2.2 Computer Vision Objectives

The goal ofComputer Vision is to understand the content of digital images. Typically, this involves developing methods that attempt to reproduce the capability of human vision. Understanding the content of digital images may involve extracting information from an image, which might be an object, a text description, a three-dimensional model, and so forth.

The goal of Computer Vision is to replicate human vision using digital images through three main processing components in consecutive order :

1. Image acquisition
2. Image processing
3. Image analysis and understanding

One of our main strengths as humans is re ected in our ability to make decisions and make sense through what we see in the real world. Providing machines and computers with this kind of visual understanding would allow them the same strength.

Object Classi cation involves the training of a model on a dataset of speci c objects, and the model classi es new objects as belonging to one or more training categories.

Object Identi cation is where a model will recognise a speci c instance of an object.

A classical application of Computer Vision is handwriting recognition for digitising handwritten content. Other methods of analysis include :

Video motion analysisuses computer vision to estimate the velocity of objects in a video, or the camera itself.

In image segmentation,algorithms partition images into multiple sets of views.

Scene reconstruction creates a 3D model of a scene inputted through images or video.

In image restoration, noise such as blurring is removed from photos using Machine Learning-based filters.[13]

FIGURE 2.1 – Human Vision vs. Computer Vision [1].

The main goals of computer vision are :

Object recognition : The field of object recognition can be broken down into a number of different visual recognition tasks and labels what appears in images.

Object recognition is a computer vision technique for identifying objects in images or videos. Object recognition is a key output of deep learning and machine learning algorithms. When humans look at a photograph or watch a video, we can readily spot people, objects, scenes, and visual details. The goal is to teach a computer to do what comes naturally to humans : to gain a level of understanding of what an image contains.

FIGURE 2.2 – Using object recognition to identify different categories of objects.

Object recognition is a key technology behind driverless cars, enabling them to recognize a stop sign or to distinguish a pedestrian from a lamppost. It is also useful in a variety of applications such as disease identi cation in bioimaging, industrial inspection, and robotic vision[14].

Segmentation Images Segmentation is  nding consistent regions in an image.

In computer vision, we are often interested to identify groups of pixels that go together. We call this problem image segmentation.Humans perform image segmentation by intuition. For instance, two people looking at the same optical illusion [15] might see different lion-visual-optical-illusionthings, all depending on how their brain segments the images. In the image below, you might see zebras, or you might see a lion.

FIGURE 2.3 – Optical illusions regarding the problem of image segmentation

One of the motivations behind image segmentation is the separation of an image into coherent objects. Here are two examples of this type of segmentation :

FIGURE 2.4 – Human segmentation of example images ; source : Svetlana Lazebnik

We might also want to segment an image into many groups based off nearby pixels being similar. We call these groups "superpixels."

Superpixels allow us to treat many individual pixels as one cluster,and therefore enable faster computations. Here is an example of an image segmented by superpixels :

FIGURE 2.5 – The superpixels allow faster computations by clustering pixels [2].

Superpixel segmentation and other forms of segmentation can help in feature support. We can treat the groups of pixels as one feature and garner information about the image from them.Image segmentation is also bene cial for some common photo effects such as background

removal. If we can properly segment an image, we will be able to only keep the groups we want and remove the others.

While segmentation is clearly useful and has many practical applications, there is no one way to segment an image, and we must compare different segmentation algorithms to nd our optimal solution. The images are prone to under-segmentation and over-segmentation if they respectively have very few or an excessive number of groups. However, even a properly segmented photo can have multiple different possible groupings.

Feature Detection: provides re ned representation of images.

Feature detection identi es key points or regions in an image that are important for further processing (e.g., matching, motion tracking). For example : detecting corners, edges, or blobs that represent important image regions [16].

Techniques Used :

SIFT (Scale-Invariant Feature Transform)

SURF (Speeded-Up Robust Features)

ORB (Oriented FAST and Rotated BRIEF)

Object tracking :

Object tracking is an application of deep learning where the program takes an initial set of object detections develops a unique identi cation for each of the initial detections and then tracks the detected objects as they move around frames in a video.

In other words, object tracking is the task of automatically identifying objects in a video and interpreting them as a set of trajectories with high accuracy.

Often, there's an indication around the object being tracked, for example, a surrounding square that follows the object, showing the user where the object is on the screen.

Video Tracking : Video tracking is an application of object tracking where moving objects are located within video information. Hence, video tracking systems can process live, real-time footage and also recorded video les. The processes used to execute video tracking tasks differ based on which type of video input is targeted. Different video

tracking applications play an important role in video analytics, in scene understanding for security and surveillance, military, transportation, and other industries. Today, a wide range of real-time computer vision and deep learning applications use video tracking methods [17].

Visual Tracking : Visual tracking or visual target-tracking is a research topic in computer vision that is applied in a large range of everyday scenarios. The goal of visual tracking is to estimate the future position of a visual target that was initialized without the availability of the rest of the video [17].

Image Tracking : Image tracking is meant for detecting two-dimensional images of interest in a given input. That image is then continuously tracked as they move in the setting. Hence, Image tracking is ideal for datasets with highly contrasting images (ex., black and white), asymmetry, few patterns, and multiple identi able differences between the image of interest and other images in the image set. Image tracking relies on computer vision to detect and augment images after image targets are predetermined [17].

Object tracking camera : Modern object-tracking methods can be applied to real-time video streams of basically any camera. Therefore, the video feed of a USB camera or an IP camera can be used to perform object tracking, by feeding the individual frames to a tracking algorithm. Frame skipping or parallelized processing are common methods to improve performance with real-time video feeds of one or multiple cameras [17].

FIGURE 2.6 – Object tracking camera [3].

3D reconstruction :

3D Computer Vision is a branch of computer science that focuses on acquiring, image processing, and analyzing three-dimensional visual data. It aims to reconstruct and understand the 3D structure of objects and scenes from two-dimensional images or video data. 3D vision techniques use information from sources like cameras or sensors to build a digital understanding of the shapes, structure, and properties of objects in a scene. This has numerous applications in robotics, augmented/virtual reality, autonomous systems, and many more.[18]

FIGURE 2.7 – 3D reconstruction

FIGURE 2.8 – 3D reconstruction

Interpretation of scenes

Scene interpretation is the process of analyzing and understanding activities in a scene by observing the motion of objects without knowing their identity, often using features like 2D position, speed, size, and binary silhouettes to track objects and detect irregularities.

## 2.3 Technological Foundations

Computer vision draws on various core disciplines and technologies, such as image processing, machine learning, and neural networks.

### 2.3.1 Image Processing

Image processing is the  rst phase of computer vision. It consists of transforming raw image data to make it analyzable. Commonly used techniques include :

Filtering : The term  ltering refers to a process that forms a new image, the pixel values of which are transformations of the original pixel values. In general, the purpose in applying  lters is the extraction of useful information (e.g., edge detection) or the adjustment of an image's visual properties (e.g., de-noising)

High-pass  ltering : remove low frequencies.

Low-pass  ltering : analogous to high-pass  lter, but remove high frequencies.

FIGURE 2.9 – Example of High-pass Image  ltering [4]

FIGURE 2.10 – Example of Low-pass Image  ltering [4]

Edge detection :The goal of edge detection is to identify sudden changes (discontinuities) in an image. Intuitively, most semantic and shape information from the image can be encoded in its edges.The edges help us extract information, recognize objects, and recover geometry and viewpoint. They arise due to discontinuities in surface normal, depth, surface color, and illumination.

An optimal edge detector must have certain qualities :

1. Good detection

   It must minimize the probability of detecting false positives (which are spurious edges, generally caused by noise) and false negatives (missing real edges, which

can be caused by smoothing, among other things). If it detects something as an edge, it should be an edge.

2. Good localization

   The detected edges must be as close as possible to the actual edges in the original image. The detector must identify where the edges occur and pinpoint the exact location of the edge; it must also be consistent in determining which pixels are involved in each edge.

3. Silent response

   It must minimize the number of local maxima around the true edge (returning one point only for each true edge point). It should tell you that there is one very speci c edge instead of splitting one edge into multiple detected edges. In other words, only the real border of the edge is captured; other possibilities are suppressed [19].

FIGURE 2.11 – Sample problems of bad edge detectors

Edge detection is extremely relevant for mammalian eyes. Certain neurons within the brain are adept at recognizing straight lines. The information from these neurons is put together in the brain for recognition of objects. In fact, edges are so useful for recognition in humans, line drawings are almost as recognizable as the original image (Fig. 8).We would like to be able to extract information, recognize objects, and recover the geometry and viewpoint of an image.

FIGURE 2.12 – Certain areas of the brain react to edges ; the line drawings are as recognizable as the original image [5].

Image transformation

Includes operations such as resizing, normalizing, and increasing data. These steps are essential to improve image quality and increase robustness of learning models [20].

Segmentation

Splits an image into separate regions or objects. Segmentation techniques include threshold segmentation, region segmentation, and convolutional neural networks for semantic segmentation [21].

### 2.3.2 Machine and Deep Learning

Machine learning, and more speci cally deep learning, has transformed computer vision. Convolutional neural networks (CNNs) have proven particularly effective for image-related tasks. These networks learn autonomously to extract meaningful features (such as shapes or textures) from large sets of labeled data.

Typical steps of a computer vision model :

Data preprocessing :This step includes resizing, normalization, and data augmentation (e.g., rotations,  ips) to improve generalization and model robustness.

Feature extraction : CNNs automatically learn spatial hierarchies of features (edges, textures, object parts) through convolutional layers, eliminating the need for manual feature design.

Classi cation or prediction : Fully connected layers or other classi ers (e.g., softmax, SVM) map extracted features to labels or predicted outputs like object categories or bounding boxes.

Post-processing :Re ning the results, such as removing false detections.

### 2.3.3 Sensors and Hardware

Computer vision relies heavily on the use of sensors, such as :

2D cameras :Capture standard images in color or grayscale.

3D cameras :Use technologies such as LIDAR or stereo vision.

Infrared cameras : Used for night vision or thermal detection.

## 2.4 Methods and Algorithms

Various algorithms and methods are used in computer vision, depending on the task.

### 2.4.1 Image Classi cation

Image classi cation categorizes and assigns class labels to groups of pixels or vectors within an image dependent on particular rules. The categorization law can be applied through one or multiple spectral or textural characterizations.

Image classi cation techniques are mainly divided into two categories :

#### 2.4.1.1 Unsupervised Classi cation

An unsupervised classi cation technique is a fully automated method that does not leverage training data. This means machine learning algorithms are used to analyze and cluster unlabeled datasets by discovering hidden patterns or data groups without the need for human intervention.

With the help of a suitable algorithm, the particular characterizations of an image are recognized systematically during the image processing stage. AI pattern recognition and image clustering are two of the most common image classi cation methods used here. Two popular algorithms used for unsupervised image classi cation are `K-means' and `ISODATA'.

K-means is an unsupervised classi cation algorithm that groups objects into k groups based on their characteristics. It is also called “clusterization.” K-means clustering is one of the simplest and very popular unsupervised machine learning algorithms.

ISODATA stands for “Iterative Self-Organizing Data Analysis Technique,” it is an unsupervised method used for image classi cation. The ISODATA approach includes iterative methods that use Euclidean distance as the similarity measure to cluster data elements into different classes. While the k-means assumes that the number of clusters is known a priori (in advance), the ISODATA algorithm allows for a different number of clusters [22].

#### 2.4.1.2 Supervised Classi cation

Supervised image classi cation methods use previously classi ed reference samples (the ground truth) to train the classi er and subsequently classify new, unknown data.

Therefore, the supervised classi cation technique is the process of visually choosing samples of training data within the image and allocating them to pre-chosen categories, including vegetation, roads, water resources, and buildings. This is done to create statistical measures to be applied to the overall image [22].

### 2.4.2 Object Detection

Object localization always starts with the process of object detection. Detection applies a deep learning model to identify potential objects within an image. Engineers utilize different techniques to detect and mark regions with objects, such as CNNs, faster R-CNN, or YOLO[23].

### 2.4.3 Semantic and Instance Segmentation

To de ne the object boundaries, some localization methods go beyond simple bounding boxes and utilize instance or semantic segmentation. Instance segmentation separates the individual object instances, while semantic segmentation assigns a predicted class to each pixel in the image [23].

FIGURE 2.13 – Instance Segmentation by Clustering Technique

### 2.4.4 Object Tracking

Object tracking, a critical component in the  eld of computer vision, refers to the process of identifying and following objects over time in video sequences. It plays a pivotal role in numerous applications, ranging from surveillance and traf c monitoring to augmented reality and sports analytics. The genesis of object tracking can be traced back to simpler times when algorithms were rudimentary and often struggled with basic motion detection in constrained environments [24].

### 2.4.5 Generation and Reconstruction

Image reconstruction is an AI-powered process central to computer vision. It involves transforming incomplete, degraded, or low-resolution images into complete, enhanced, or high-resolution versions. Image reconstruction serves an important role in  elds like medical imaging, satellite imagery, and digital forensics. It excels in areas where the clarity and detail of an image can signi cantly in uence outcomes.

Image reconstruction evolved from basic interpolation techniques. This involves estimating missing pixels by averaging the values of surrounding pixels. In turn, this evolved into more sophisticated methods using machine learning and deep learning.

Following that, the development of Convolutional Neural Networks (CNNs) was a watershed moment in the  eld. CNNs are adept at capturing spatial hierarchies in images. This makes them ideal for high-resolution image reconstruction from their lower-resolution counterparts.

The introduction of the Super-Resolution Convolutional Neural Network (SRCNN) later demonstrated that deep learning models could outperform traditional image resolution methods [25].

FIGURE 2.14 – Comparisons of bicubic-upsampled, ESRGAN, RealSR, and Real-ESRGAN results on real-life images

Finally, Generative Adversarial Networks (GANs) pushed image reconstruction further by generating new image pixels with unprecedented realism. Specialized models like ESRGAN

(Enhanced Super-Resolution Generative Adversarial Networks) set new standards for high-resolution outputs.

## 2.5 Applications of Computer Vision

Computer vision  nds applications in almost every  eld, revolutionizing the way machines perceive and interact with their environment.

### 2.5.1 Manufacturing

Computer vision monitors equipment for maintenance and checks packaging and product quality on production lines.

### 2.5.2 Healthcare

Medical imaging systems can assist doctors in making faster and more accurate diagnoses.

### 2.5.3 Transportation

Computer vision systems are being increasingly applied to increase transportation ef ciency. For instance :

Detection of traf c signal violators, allowing law enforcement agencies to minimize unsafe on-road behavior

Intelligent sensing solutions for detecting speeding and wrong-side driving violations

Traf c  ow analysis in intelligent transportation systems

### 2.5.4 Trade

Today, tech giants such as Amazon are actively exploring how retail can be revolutionized using AI vision :

“Take and leave” shopping experiences

Monitoring shopper activity for loss prevention

Analyzing customer moods to personalize advertisements

### 2.5.5 Security and Surveillance

Computer vision enables :

Real-time monitoring

Facial recognition

Behavior analysis in public and private spaces

Enhancing threat detection and response times.

### 2.5.6 Agriculture

Crop monitoring, weed detection, and nutrient de ciency analysis can be done using drone, satellite, or aerial imagery.

## 2.6 Challenges and Perspectives :

Challenges include handling varying lighting conditions, occlusions, and the need for large annotated datasets ; however, advancements in AI continue to expand its accuracy and adaptability.

Computer Vision technology has ushered in a transformative era, rede ning the way industries operate and introducing unprecedented possibilities. Its capacity to analyze and interpret visual data has ignited innovation across sectors such as healthcare, automotive, manufacturing, and beyond. From enabling autonomous vehicles to improving medical diagnostics, Computer vision has indeed broadened the horizons of what's achievable.

However, numerous challenges lie ahead [26].

Optimizing AI-Based Computer Vision for Mobile Robots

In the AI-based computer vision algorithms era, running some of the computation-heavy algorithms without a super-computer is challenging. Making the neural networks model t on a limited HW built into a mobile robot is challenging.

Detecting Unusual and Deformed Objects

Computer vision algorithms are an essential part of all autonomous vehicles. Currently, they do pretty well in standard road traf c scenarios. What poses a challenge is identifying unusual methods like crashed cars, etc. Generally speaking, all sorts of deformations are a challenge to object detection algorithms since it's dif cult to train such a network model with virtually unlimited possibilities of deformed or damaged object images.

Object Occlusion in 3D Scene Reconstruction

Object occlusions are still a challenge for computer vision algorithms that try to reconstruct a 3D scene from a series of images. Humans have no problem identifying that two visible parts of a dog standing behind a tree are still the same. Unfortunately, this is not the case for computer vision algorithms.

Dynamic Lighting in the Wild

Working in dynamic lighting conditions has always been challenging for computer vision. Out there in the wild, with shadows and re ections on the objects, it's often hard to correctly identify an entity or asses it's orientation.

## 2.7 Future Prospects :

The future of CV lies in autonomous systems, smart cities, and augmented reality, with ongoing research pushing toward more human-like visual understanding and real-time decision-makin

One of these technologies :

Enhanced AI Models and Algorithms

Advanced Hardware

Edge Computing and 5G Integration

Improved Data Availability and Quality

Emerging Technologies and Innovations

Key Industries and Application Areas

Autonomous Driving and Mobility

Smart Cities and Public Safety

## 2.8 Conclusion

Among these applications, automated access management, such as that deployed in the University of Saida park, demonstrates the ability of computer vision to enhance security and ef ciency in speci c contexts. This solution, which relies on technologies such as license plate recognition and object detection, will be analyzed in depth in the next chapter.

CHAPTER

# 3

# Application of Computer Vision to Access Management at the University of Saida Parking

## 3.1 Introduction

Managing access to the University of Saida parking lot poses a major logistical challenge due to the growing number of vehicles and security constraints. The integration of computer vision, particularly through the YOLO (You Only Look Once) algorithm, makes it possible to automate license plate recognition and ef ciently control access. This chapter details the speci cations, the application of YOLO, the preparation of the datasets, the tests conducted, as well as the limitations and prospects of this system.

## 3.2 Speci cations

The parking access management system must meet the following requirements :

Functional Objectives :

Detect vehicles entering and exiting the parking lot.

Automatically identify the license plates of vehicles entering and exiting the parking lot.

Verify access authorization by comparing the plates with a database of registered vehicles.

Alert in case of unauthorized access attempts.

Technical constraints :

Use of high-resolution cameras to capture license plates in various conditions (day, night, bad weather).

Real-time processing to minimize access delays.

FIGURE 3.1 – Access of the university of Saida

FIGURE 3.2 – Access's barriers of the university.

## 3.3 Application of YOLO

YOLO, an object detection algorithm based on convolutional neural networks (CNNs), is used to detect and read license plates. Implementation steps include :

Version selection :YOLOv12 is chosen for its speed and accuracy, suitable for real-time processing.

YOLO12 introduces an attention-centric architecture that departs from the traditional CNN-based approaches used in previous YOLO models, while maintaining the real-time inference speed essential for many applications. This model achieves industry-leading accuracy in object detection through methodological innovations in attention mechanisms and the overall network architecture, while maintaining real-time performance [27].

Key Features :

Area Attention Mechanism : A new approach to self-attention that ef ciently handles large receptive  elds. It divides feature maps into equal-sized regions (4 by default), either horizontally or vertically, avoiding complex operations and maintaining a large effective receptive  eld. This signi cantly reduces computational costs compared to standard self-attention.

Residual Ef cient Layer Aggregation Networks (R-ELAN) : An enhanced feature aggregation module based on ELAN, designed to address optimization challenges, particularly in large-scale attention-centric models. R-ELAN introduces :

Block-level residual connections with scaling (similar to layer scaling).

A redesigned feature aggregation method creating a bottleneck-like structure.

Optimized Attention Architecture : YOLO12 streamlines the standard attention mechanism for greater ef ciency and compatibility with the YOLO framework. This includes :

Using FlashAttention to minimize memory access costs.

Removing positional encoding for a cleaner and faster model.

Adjusting the MLP ratio (from 4 to 1.2 or 2) to better balance computation between the attention and feedforward layers.

Reducing the stacked block depth for better optimization.

Leveraging convolution operations (where available) for their computational ef ciency.

Adding a 7x7 separable convolution (the "position perceiver") to the attention mechanism to implicitly encode position information.

Comprehensive Task Support: YOLO12 supports a range of essential computer vision tasks : object detection, instance segmentation, image classi cation, pose estimation, and oriented object detection (OBD).

Increased Ef ciency : Achieves higher accuracy with fewer parameters compared to many previous models, demonstrating a better balance between speed and accuracy.

Flexible Deployment: Designed for deployment on a variety of platforms, from edge devices to cloud infrastructure.

FIGURE 3.3 – Comparison of heat maps between YOLOv10, YOLOv11,and the proposed YOLOv12. Compared to the advance YOLOv10 and YOLOv11, YOLOv12 demonstrate a clearer perception of objects in the image. All the results are obtained using the X scale models. Zoom into compare the details.

FIGURE 3.4 – Detection performance. (COCO val2017)

Processing pipeline:

Image acquisition : A camera captures the image of the vehicle at the entrance.

License plate detection : YOLOv12 locates the license plate in the image.

Segmentation and recognition : Once the license plate is detected, a Tesseract-based OCR (Optical Character Recognition) module extracts the alphanumeric characters.

Veri cation : The extracted data is compared to a database.

Action : If the license plate is authorized, the barrier opens; otherwise, an alert is sent.

### 3.3.1 Functional diagram

FIGURE 3.5 – Functional diagram of the project.

Camera : Captures the image in real time.

Preprocessing : Brightness adjustment, noise reduction.

YOLOv12 : Detection.

Barrier/Alert : Final action based on the result.

### 3.3.2 Diagram with recognition of plates

Vehicle Number Plate Recognition (NPR) or License Plate Recognition (LPR) or Registration Plate Recognition (RPR) is an enhanced computer vision technology that connects vehicles without direct human connection through their number plates.

FIGURE 3.6 – The steps of an RPR system.

FIGURE 3.7 – Neural network-based parking system real-time license plate detection

FIGURE 3.8 – Diagram with recognition of plates.

### 3.3.3 Dataset preparation :

Dataset quality is crucial to YOLO's performance. The preparation steps include :

Data collection :

Image capture.

Variety of condition : Day, night, rain, various angles.

FIGURE 3.9 – dataset preparation.

Annotation :

Use of LabelImg.

Creation of annotation  les in YOLO format (bounding box coordinates and classes).

FIGURE 3.10 – Dataset label Distribution.

FIGURE 3.11 – Con dence curve.

FIGURE 3.12 – Annotation (Labelling with YOLO)

Increase in data :

Application of transformations (rotation, blur, change in brightness) to increase the robustness of the model.

FIGURE 3.13 – Application of transformations.

Division : 80% for training, 10% for validation, 10% for tests.

FIGURE 3.14 – Division of dataset result.

Storage : images and annotations are organized in a structure compatible with YOLOV12.

FIGURE 3.15 – Storage of result.

## 3.4 Trials

The tests were carried out on a prototype installed at the entrance of the main car park :

FIGURE 3.16 – Trial in real time (Personal view).

FIGURE 3.17 – Trial in real time (System view).

Results :

Plate detection accuracy : 80% of the results

Average processing time : 3 vehicle per second.

Correct character recognition rate : 65%

Authorized vehicle access : quick opening of barrier.

Problems observed :

Detection errors in low light.

Confusion with damaged or partially masked plates.

High-res video requires compression, losing detail.

Prevent compact devices from running high-end AI models.

## 3.5 Limits and prospects

### 3.5.1 Limits :

Real-time systems face inherent constraints including architectural in exibility due to specialized hardware/software requirements [28], signi cantly higher development costs from rigorous temporal veri cation [29], and limited computational complexity as they prioritize predictable WCET over advanced analytics [30]. These systems also struggle with scalability from  xed-priority designs [28] and unique vulnerabilities to timing-based security threats [31]. As noted in research, "static priority assignment creates fundamental scalability limits" [28] while "denial-of-service attacks targeting temporal behavior represent a unique threat vector" [31], demonstrating the multifaceted challenges in real-time system design.

### 3.5.2 Prospects :

The advancement of Real-Time Systems (RTS) is being transformed by several key technologica developments. In Edge Computing and Internet of Things (IoT) applications, RTS enable sub-millisecond latency processing while reducing cloud dependence by 60-80% for time-sensiti operations [32]. The integration of Arti cial Intelligence and Machine Learning (AI/ML) has become particularly impactful, with quantized neural networks achieving <10ms inference times on Real-Time Operating System (RTOS) platforms [33]. Fifth-Generation (5G) and Sixth-Generation (6G) cellular technologies further enhance these capabilities through network slicing that guarantees temporal isolation [34]. These developments are critical for autonomous systems, where Robot Operating System 2 (ROS 2) extensions

now provide deterministic execution for navigation tasks [35]. In healthcare, real-time electroencephalogram (EEG) processing detects neurological events 300ms faster than human assessment [36]. Emerging Quantum Computing (QC) architectures suggest future breakthroughs, particularly for complex scheduling problems in hybrid quantum-classical systems [37].

## 3.6 Conclusion

The application of computer vision, via YOLOv12, to parking access management at the University of Saida offers a modern and ef cient solution. Tests demonstrated satisfactory performance, although improvements are needed to overcome environmental and technical limitations. This project paves the way for similar applications in other university infrastructures, with signi cant optimization and expansion potential.

CHAPTER

# 4

# Development of the university access control prototype using computer vision

## 4.1 Presentation of the prototype

The prototype created is a scaled-down model that simulates an automated vehicle access control system at the university entrance. The objective is to demonstrate the functioning of the access control system using a physical model and embedded technologies. Composition of the prototype :

Miniature cars : Used to simulate the vehicles of university members.

Road model : A reproduction of a university access road with ground markings.

Automatic barrier : Made with a servo motor that raises or lowers the barrier based on the detection result.

Traf c light : A signaling system (red, orange, green) controlled by an ESP8266 to regulate access.

CAM : Camera module that captures real-time images of cars at the entrance for license plate detection or visual recognition.

Communication system : All modules (ESP8266 + CAM) are communicate with a PC that runs the YOLOv12 detection algorithm.

FIGURE 4.1 – Photo of the prototype showing the miniature road, the barrier, the traf c light, the camera, and the miniature cars.

### 4.1.1 Flowchart

The diagram below illustrates how the access control system works :

FIGURE 4.2 – Prototype operation diagram

The PC runs YOLOv12 for license plate detection or vehicle recognition.

CAM module captures images and sends them to the PC via USB.

The ESP8266 receives decisions from the PC (access granted or not) and :

Activates the servo motor to raise or lower the barrier.

Controls the traf c light to signal the status (red, orange, green).

| Components | Description / Role |
|---|---|
| CAM | Camera module connected to the PC via USB |
| ESP8266 (NodeMCU) | Control the servo and the LEDs |
| Servo moteur | Automatic barrier |
| 3 x LED (R, O, V) | Traf c light |
| Resistances (330) | For the LEDs |
| Breadboard | Physical connections |
| Alimentation 5V | (via USB or battery) |

TABLE 4.1 – Components of prototype

### 4.1.2 Simpli ed electrical diagram

#### 4.1.2.1 Objective of the simpli ed diagram

Show the connections between the following components :

CAM (Elecom Ucam-dln130tpn)

ESP8266

Servo motor (for the barrier)

LED (traf c light : red, orange, green)

Power supply

#### 4.1.2.2 Components to place in the schematic (Fritzing or Proteus) :

a) CAM (Elecom Ucam-dln130tpn)

A-Description : an Elecom UCAM-DLN130T series webcam in a light pink color. It features a cylindrical camera unit that rests on a foldable base. The camera unit has "HD 1280 x 720p" printed on it, indicating its resolution capabilities [38].

FIGURE 4.3 – CAM (Elecom Ucam-dln130tpn)

b) NodeMCUESP8266

A-Description : Is a more powerful microcontroller. A SOC often has a powerful processor (CPU) and a GPU. A SOC is typically built around a microcontroller or microprocessor. SOC can be found in mobile phones, microwaves, robots, hospital equipment and cars. Namely, the Node MCU is based on the ESP8266 SoC and is signi cantly more powerful than the Arduino. It is exactly like the Arduino, except that the only thing that is more integrated is the Wi-Fi. It also gives you options in terms of programming languages. You can either program it in Lua or APL "Arduino Programming Language". Note that it is recommended for beginners to start with APL, unless the implementation requires otherwise. The Node MCU brings much more processing power, allowing you to build just about anything from smart appliances to autonomous robots. And it's compatible with pretty much everything the Arduino is, and sells in the same price range [39].

FIGURE 4.4 – ESP8266 Card (NodeMCU)

FIGURE 4.5 – Pinout of the NodeMCU ESP8266

B-Features This module is primarily based on the ESP-12E version of the ESP8266, with however some speci cities that are unique to it [40] :

he module's interface is primarily divided into two parts, including the  rmware and hardware, where the  rmware execution takes place on the ESP8266 SoC Wi-Fi and the hardware is based on the ESP-12 module.

The  rmware is based on Lua, an easy-to-learn scripting language, offering a simple programming environment and a fast scripting language that you connects you to a large community of developers.

The rmware being open source gives you the ability to edit, modify, and rebuild the existing module and constantly change the interface until you can optimize the module according to your needs.

The USB to UART converter is added to the module, thus facilitating the conversion of USB data to UART data, which primarily follows a serial communication language.

Instead of the usual USB port, a MicroUSB port is included in the module to connect it to the computer, serving a dual purpose : programming and powering the board.

The board includes an LED indicator that blinks and turns off instantly, indicating the current status of the module if it is functioning correctly when connected to the computer. (You may need to install some drivers on your computer if it fails to detect the NodeMCU board)

The module's ability to establish a perfect WiFi connection between two channels makes it an ideal choice for integrating with other embedded devices such as the Raspberry Pi.

c) Servomotor :

A servomotor (Figure 4.6) is an actuator capable of maintaining static resistance while continuously monitoring and adjusting its position based on feedback. The servomotor consists of several visible components :

Horns (control arms).

A rotational shaft that supports a plastic or metal accessory.

A protective casing

Additionally, it contains internal components housed within the casing :

A DC motor.

Gear mechanisms forming a reduction gearbox.

A position sensor (often a potentiometer) to measure the shaft's orientation angle.

An electronic control board for regulating shaft position and driving the DC motor.

The position is de ned with an angle travel limit of 180 degrees. Often abbreviated to “servo” from the Latin servus meaning “slave”. There are several servomotors, in our application we used a low power (9 Gram) SG90 servo motor. It contains three wires, one for power supply, the other for ground (GND) and the last for input signal (control) [41].

FIGURE 4.6 – View of a servo motor

B- Characteristics [42]

j- Pin numbers : 3 wires (Brown : Mass) (Red : Vcc) (Orange : Command)
k- Dimension : 22mm × 11.5mm × 27mm
l- Weight : 9 grams
m- Min supply voltage : 4.8V
n- Max supply voltage : 6V
o- Speed : 0.12 s/60° under 4.8V
p- Couple : 1.2 kg/cm under 4.8V
q- Consumption : 125 mA

d) LED's

LED is the acronym for "Light Emitting Diode" in English, which translates into French as "diode électroluminescente". A LED is an electronic component capable of producing light when an electric current passes through it. Unlike traditional bulbs, LEDs are more energy ef cient, more durable and offer a wide variety of colors. They are used in many areas such as lighting, display screens, road signs and electronic devices.

FIGURE 4.7 – LED

e) Breadboard :

The experimental board or prototype board in Figure 4.1, also called breadboard, is a device that can produce prototypes of electronic circuits and test them.

FIGURE 4.8 – Bread Board

The simpli ed schematic in FRITZING :

FIGURE 4.9 – Simpli ed electrical diagram of the prototype

## 4.2 Technical details of the codes

### 4.2.1 ESP8266 Code

This code manages the barrier (servo) and the traf c light based on commands received from the PC via Wi-Fi.

FIGURE 4.10 – ESP8266 Code operation diagram

The ESP8266 listens on port ( ) for messages from the PC.

If the message is "authorized," it turns on the green light and opens the barrier.

Otherwise, red light and barrier closed.

The code is simple, clear, and adapted to a mockup. (Complete code in 4.2)

### 4.2.2 Code Python

(complete code in 4.1)

FIGURE 4.11 – Python Code diagram

#### 4.2.2.1 Code Python (of license plate recognition)

FIGURE 4.12 – Python Code diagram (License Plate Detection)

## 4.3 Preparation of the datasets

To train the YOLOv12 model to recognize toy cars :

Data collection : Photos of the miniature cars were taken from various angles, in different positions and lighting, to create a dataset representative of real-life situations.

FIGURE 4.13 – Data collection

Annotation : Each car in the photos has been annotated with bounding boxes using a tool like LabelImg, in YOLO format.

FIGURE 4.14 – Annotation (Labelling)

FIGURE 4.15 – Dataset Label Distribution

FIGURE 4.16 – Dataset Label Correlogram/Pair Plot

These two  gures are visualizations of the bounding box annotations from a dataset, likely created or analyzed using Ultralytics YOLO tools (which now extend beyond v8, so v12 is a plausible future iteration or custom naming). These plots help in understanding the distribution and characteristics of the objects labeled within your dataset. Figure 4.15 provides four key insights into your YOLOv12 dataset's annotations. Figure 4.16, often called a "pair plot" or "scatterplot matrix," shows the relationships between pairs of variables (x, y, width, height) and their individual distributions. It's particularly useful for seeing correlations and value ranges.

Cleaning : Deletion of unusable images (blurry or poorly framed).

FIGURE 4.17 – Cleaning methode

Augmentation : Application of rotations, zooms, brightness changes, and other augmentation techniques to increase the diversity of the dataset.

FIGURE 4.18 – Augmentation Application

## 4.4 Learning

The YOLOv12 model has been adapted to recognize miniature cars, which differs slightly from a standard model intended for real vehicles.

Architecture : YOLOv12, designed for rapid object detection, has been con gured to primarily detect one class : miniature car.

Training : Conducted on a PC with GPU, using the custom dataset. The model was trained until performance stabilized.

Parameters : Image size 416x416, adjusted learning rate, and testing over multiple epochs to avoid over tting.

Validation : A portion of the dataset was used for validation. The model achieved a detection accuracy of 96% across all tests.

FIGURE 4.19 – The training results

FIGURE 4.20 – Results training with Yolov12

FIGURE 4.21 – File of results training with Yolov12 illustrate the number of epoch used, Time, Box loss, Precision. . . . . .ect.

These images, displays a standard set of training metrics plots generated by Ultralytics YOLO after a model training session. These plots are crucial for evaluating the training process and the performance of your object detection model over epochs. The grid consists of 9 subplots, showing various loss functions and performance metrics for both the training and validation datasets across 100 epochs. The plots demonstrate that your YOLOv12 model's training process was highly successful and stable over 100 epochs. These results are indicative of a well-trained model that should perform effectively for its intended object detection and tracking tasks in a controlled environment and likely generalize well to similar real-world conditions if the training data were representative.

FIGURE 4.22 – Precision-con dence Curve.

This plot, titled "Precision-Con dence Curve", is a critical visualization for evaluating the trade-off between the precision of your object detection model and the con dence threshold applied to its predictions. It shows how the model's precision (the proportion of correct positive predictions among all positive predictions) changes as you vary the minimum con dence score required for a detection to be considered valid.

## 4.5 Test and trial

### 4.5.1 Laboratory tests :

The system was tested with different miniature cars and under various lighting conditions. YOLOv12 detected the cars in less than 150 ms per image.

YOLOv12 detected cars in less than 150 ms per image.

Communication between the CAM and the PC was smooth via USB, with real-time image transmission. The communication between the CAM and the PC was smooth via USB, with real-time image transmission.

### 4.5.2 Practical tests :

When a toy car approaches :

The CAM camera captures the image.

The image is sent to the PC.

YOLOv12 detects the car.

The barrier is lifted (command sent to the ESP8266) and the light turns green.

Otherwise, the light stays red and the barrier does not open.

Reliability : The system has demonstrated good robustness, even during more intense simulated traf c (several cars arriving one after the other).