Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Hassiba Benbouali de Chlef

Faculté des Sciences Exactes & Informatique

Département d'Informatique

Mémoire

Présenté pour l'obtention du diplôme de

LICENCE EN INFORMATIQUE

Spécialité : Systèmes Informatiques

Par

Ilyas Ali Benyahia

Ayyoub Allali

Thème :

License Plate detection and recognition with OpenCV and C++

Soutenue le 25/06/2023 devant le jury composé de :

Bechar Rachid	MCA/ UHB-Chlef	Président
Boukhelef Djillali	MAA/ UHB-Chlef	Examinateur
Guellil Zouaoui	MAA/ UHB-Chlef	Encadreur

Année Universitaire 2022-2023

Dedications

“

I dedicate this humble work

I start by thanking my dear father who sacrificed for me. It wasn’t just your contribution to my education, but you have also been my support in life. I am proud that you are my father and I thank you for All you do for me.

And I also thank my beloved mother, who was Support since my childhood. She gave me constant love and Encouragement and I owe it to her for her success today. Thank you, dear mom. I cannot forget the generous family that was the best support for me

Finally, I extend my thanks and appreciation to Brothers and sisters, and to all my friends on Sympathy, friendship and solidarity with me. I hope you He will accept my greetings and appreciation.

”

- Ayyoub - Ilyas -

Acknowledgment

“

First of all, I thank ***Allah*** *the Almighty for giving me the courage and The patience necessary to complete this work. I would especially like to thank my supervisor****“Mr. GUELLIL ZOUAOUI ”*** *who was the cornerstone of my My educational career. Best regards and deep thanks for your valuable guidance. You inspire me It greatly contributed to the development of my abilities and achieving my goals. I am infinitely grateful to you,*

*I hope that the members of the jury will find “****Bechar Rachid*** *” and “****Boukhelef Djillali*** *” here an expression for me Sincere thanks for the honor they have given me by taking It’s time to read and evaluate this work. Our thanks also go to the faculty and administrative staff of Faculty of Exact Sciences at the University of Chlef for its richness and the quality of their education and those who make great efforts to provide it Their students have modern and excellent training. I would like to extend my thanks to my family and friends for Their eternal support and encouragement duringthese difficult years of life education Finally, I would like to thank everyone who contributed directly or indirectly to accomplish this work..*

”

ملخص

الكشف عن لوحة الترخيص والتعرف عليها هي تكنولوجيا تعتمدعلى رؤية الكمبيوتروالتعلم الالي ومن بين الاشكاليات هذه الدراسةهي اكتشاف وتعرف دقيق على لوحة الترخيص في ظروف مختلفة، ومعالجة مخاوف الخصوصية والأمان. يهدف هذا المشروع الى إنشاء نظام قوي وفعال قادر على اكتشاف لوحات الترخيص والتعرف عليها بدقة. وقد تم استخدم مكتبات opencv وtesseract_OCR بلغة برمجة C++ داخل بيئة Visual studio ولقد تضمن هذا البحث عدة مراحل من بينها فتح الكاميرا واكتشاف اللوحات والتعرف على الأحرف الموجودة فيها . تم تصميم النظام لتوفير نتائج في الوقت الفعلي، مما يدل على كفاءته وفعاليته.

كلمات مفتاحية:

OpenCV، رؤية الكمبيوتر ، الكشف ، التعرف ، لغة البرمجة C ++، Tesseract OCR

Abstract

License plate detection and recognition is a technology based on computer vision and machine learning. Among the problems of this study is accurate detection and recognition of license plate in different conditions, addressing privacy and security concerns. This project aims to create a robust and effective system capable of detecting and recognizing license plates accurately. The OpenCV and OCR Tesseract libraries were used in the C++ programming language within the Visual Studio environment. This research included several stages, including opening the camera, discovering the paintings, and identifying the letters in them. The system is designed to provide real-time results, demonstratingits efficiency and effectiveness.

Keywords : OpenCV, computer vision, detectionrecognition,C++ programming language,Tesseract OCR.

Résumé

La détection et la reconnaissance des plaques d’immatriculation est une technologie basée sur la vision par ordinateur et l’apprentissage automatique, des défis tels que la détection et la reconnaissance des plaques dans diverses circonstances, la protection de la vie privée et de la sécurité. Les bibliothèques OpenCV et Tesseract OCR ont été utilisées et l’application a été implémentée à l’aide du langage C++ dans l’environnement Visual Studio. La phase de mise en œuvre tournait autour du processus de détection et de reconnaissance de la plaque, et comprenait l’importation des bibliothèques nécessaires, l’ouverture de la caméra, la découverte des plaques et l’analyse des caractères dessus. Le système est conçu pour fournir des résultats immédiats, démontrant son efficacité et son efficience.

Mots clés : OpenCV, vision par ordinateur, détection, reconnaissance, langage de programmation C++, Tesseract OCR.

Contents

Dedications . . . . **I**

Acknowledgment . . . . **II**

General Introduction **1**

1 Computer Vision **4**
- 1.1 Introduction 5
- 1.2 Definition Of Computer Vision 5
- 1.3 Purpose Of Computer Vision 6
- 1.4 Advantages Of Computer Vision 6
.
- 1.5 Images 6
  - 1.5.1 Definition Of An Image 6
  - 1.5.2 Image Representation 7
    - 1.5.2.1 Image In Pixels (Raster) 7
    - 1.5.2.2 Vector Image 8
  - 1.5.3 Characteristics Of An Image 8
  - 1.5.4 Digital Image Interpretation 10
    - 1.5.4.1 Digital Image Color Coding 10
    - 1.5.4.2 Cumulative Histogram Of An Image : 11
  - 1.5.5 Digital Image Processing 12
    - 1.5.5.1 Definition Of Image Processing 12
    - 1.5.5.2 Segmentation D'Image 12
    - 1.5.5.3 Filtering 13
- 1.6 Computer Vision Applications 13
- 1.7 Conclusion 14

2 Presentation Of The OpenCv Libraries. **15**
- 2.1 Introduction 16
- 2.2 History 16
- 2.3 What Is OpenCV Library ? 17
  - 2.3.1 Image Processing 18
  - 2.3.2 Video Processing 19
- 2.4 OpenCV Modules 19
- 2.5 Structure And Content Of Library OpenCV 20
- 2.6 Using OpenCv 22
- 2.7 Conclusion 23

3 Conception Et Implémentation 24
3.1 Introduction 25
3.2 Problematic 25
3.3 Working Environment 25
3.3.1 Hardware Environment 26
3.3.2 Software Environment 26
3.4 Steps To Detect And Recognize License Plate 29
3.4.1 Step 1 : Import Libraries 29
3.4.2 Step 2: Open Camra 30
3.4.3 Step 3 - Starte To Detect License Plate : 30
3.4.4 Step 4 - Recognition Number Plates : 31
3.5 Implementation 32
3.6 Work Result 34
3.7 Conclusion : 34

General Conclusion 35

Bibliography or reference 36

List Of Figures

1.1 Different types of images 7
1.2 Raster image. 7
1.3 Raster Image and Vector Image. 8
1.4 20x20 size image. 8
1.5 Example illustrating different image resolutions 9
1.6 Example of an image affected by noise 9
1.7 binary image. 10
1.8 Grayscale image 11
1.9 color image. 11
1.10 Illustration of image segmentation (figure1: original image, figure2: segmented image) 12
1.11 Example of a filter on an image. 13

2.1 timeline of opencv. 17
2.2 Logo OpenCV. 18
2.3 Module of OpenCV. 20
2.4 Structure of OpenCV. 22

3.1 Create a new project. 27
3.2 Additional Include Directories. 27
3.3 Additional Library Directories. 28
3.4 Add Dependencies. 28
3.5 Add opencv_world412d.lib. 28
3.6 application interface 33
3.7 How the app works. 33
3.8 application interface 34

General Introduction

Context

Ensuring the safety of roads and drivers has always been one of the main tasks of various organizations dealing with road network management and safety. This need requires the implementation of systems such as radars, roadblocks, and checkpoints, and with the expansion of the automotive market, managing this huge network becomes very difficult. Conventional means may seem ineffective when they require accurate identification and differentiation of vehicles, such as stolen vehicle detection and license plate theft.

Problem

Within computer vision, license plate detection and recognition is a challenging issue. Accurate detection and identification of license plates from images or video streams are essential for applications such as monitoring, traffic management, and security. However, differences in lighting conditions, complex backgrounds, and different license plate designs present significant challenges to current methods.

Making up for the shortcomings of the currently available methods requires taking advantage of innovative ideas that are simple, fast, and easy to implement. In this context, we have developed an efficient and practical solution. Our solution is by leveraging com- puter vision technologies and harnessing the power of the OpenCV library together with the C++ programming language.

Objectives

The project aims to develop an effective license plate detection and recognition system. This will contribute to streamlining traffic management and identifying suspicious drivers and vehicles. Thanks to this innovative approach, we will contribute to improving road safety and enhancing security in urban and rural transport environments.

Memory Organization

This project is organized into several chapters :

Chapter 1: Computer Vision

- Provides an introduction to the concept of computer vision, its definition, and its purpose.

- Discusses the advantages of using computer vision techniques.

- Explains the characteristics of images, digital image interpretation, and digital image processing.

- Presents various applications of computer vision.

Chapter 2: Presentation of the OpenCV Library

- Introduces the OpenCV library and its history.

- Explains the structure and content of the OpenCV library, including its modules.

- Discusses how to use OpenCV in the project.

Chapter 3: Conception and Implementation

- addresses the project's problematic aspects Introduces the project's design and implementation phase.

- Describes the working environment, including the hardware and software requirements.

- Outlines the steps involved in detecting and recognizing license plates.

- Presents the implementation details.

- Discusses the achieved results of the project.

Chapitre 1

Computer Vision

1.1 Introduction

With the advancement of technology and the development of computers, computer vision has become a vital field in multiple fields. This field relates to the ability to analyze and understand images and videos by computers, with the aim of extracting information and making decisions based on visual content.

One of the common applications of computer vision lies in the recognition and detection of number plates. When we talk about number plates, we are referring to the plates used to identify vehicles, whether they are cars, motorcycles or buses.

Recognizing and detecting number plates requires analyzing the image captured by the camera, distinguishing the letters and numbers on the plate and extracting them accurately. This is done by image processing and machine learning algorithms that enablethe computer to recognize the patterns and structure of letters and numbers.

Computer vision technologies can be used in many areas ranging from security, traffic management, improving user experience in image inference applications, and more. For example, number plate recognition can be used in traffic control systems to improve safety and monitor traffic violations, or in vehicle entry and exit systems to facilitate identification and access regulation.

1.2 Definition Of Computer Vision

Computer vision is a field of artificial intelligence and computer science that focuses on enabling machines to interpret, analyze, and understand visual information from the real world and provide accurate results.

Computer vision aims to develop different algorithms and technologies that can help computers process and analyze images, videos, and other visual data like human vision. It also works on object detection and recognition, image segmentation, tracking, motion analysis, image recovery, and image enhancement.

We can distinguish two levels of image processing used by computer vision:

- High-level vision, which is concerned with recognizing and interpreting complex visual information such as objects, scenes, and events (eg recognizing handwritten letters).

- Low-level vision that is concerned with processing and analyzing basic information related to images and visual scenes, such as surfaces, colors, lighting, image filtering, edge detection, etc.

In general, computer vision has the potential to revolutionize various fields and has become an essential area of research and development

1.3 Purpose Of Computer Vision

Computer vision consists of reproducing the results obtained by human vision on a computer using computer means by replacing the eye with a camera and the brain with a computer.

A computer vision computer system uses as input one or more digital images acquired using a camera. These images undergo a set of processing to extract the maximum amount of information relating to the scene to be recognized.

1.4 Advantages Of Computer Vision

Computer vision provides many advantages and benefits in several areas, the most important of which are: [1]

- **High accuracy :** Computer vision has the ability to detect and analyze with high accuracy, and this can help to obtain accurate and correct results in fields such as medical diagnosis and remote sensing.
- **The ability to deal with large amounts of data :** computer vision deals with large amounts of data, and this makes it able to deal with information in an efficient and regular manner.
- **Speed :**Computer vision can analyze and process images and videos very quickly, which helps to get results quickly.
- **High ability to recognize and detect:** Modern technologies in computer vision such as artificial neuron networks and deep learning are used to identify objects and objects in images and videos with high accuracy, which enables them to be used in applications such as face recognition and license plate recognition.
- **Save time and costs:** Relying on technology in computer vision helps save time and costs in areas such as product inspection in industry and X-ray inspection in the medical field.

1.5 Images

1.5.1 Definition Of An Image

Image is a visual representation of a real-world scene, obtained by projecting a group of light rays. An image is made up of a group of pixels, which are the smallest units that make up an image. Each pixel is characterized by two coordinates (X, Y) that represent its position in the image.

From a human point of view, it contains multiple semantic information, such as shapes, colors, and patterns, which allow humans to understand and interpret the content of the image. And from a mathematical point of view, an image is represented by an array of numbers that correspond to the value of each pixel in the image.

There are three main types of images as displayed in figure 1.1:

- Binary images : These images contain only two values in each pixel, either 0 or 1
- Grayscale images.
- Color images.

FIG. 1.1: Different types of images

1.5.2 Image Representation

We can distinguish two types of images, the image in pixels (or matrix) and the Vector image.

1.5.2.1 Image In Pixels (Raster)

A pixel image is made up of a grid of pixels, with each pixel containing a specific color or grayscale value. Pixels are represented in the form of small squares. Pixel images are used for digital images, screenshots, scanned images, etc.

Image resolution refers to the total number of pixels in an image, which is usually given as the width and height of the image in pixels. For example, a 1920x1080 image would have 1,920 pixels in width and 1,080 pixels in height, resulting in a total of 2,073,600 pixels in the image. The more pixels an image contains, the higher its resolution and the more detail it can display.as displayed in figure 1.2

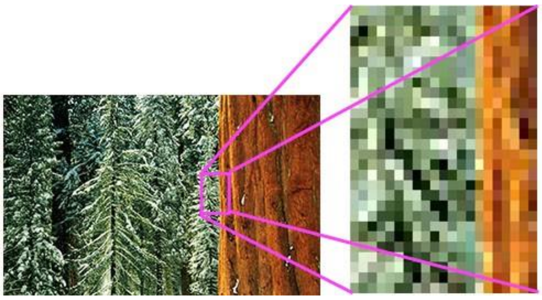

FIG. 1.2: Raster image.

1.5.2.2 Vector Image

A vector image is devoid of matrices and consists of mathematically defined lines, shapes, and colors, rather than pixels.

In other words, it is composed of individual geometric objects (line segments, polygons, arcs, etc.). Vector images can be resized without losing quality because they are based on mathematical formulas, unlike an image in pixels.[2]

Here is an example of a comparison between Raster Image and Vector Images as displayed in figure 1.3

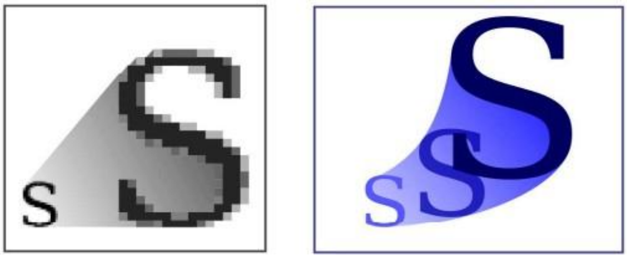

FIG. 1.3: Raster Image and Vector Image.

1.5.3 Characteristics Of An Image

the image is an information structure set characterized by the following parameters :

- **Dimension :**

  Image dimensions refer to their physical size, measured in pixels, whose elements are numeric values in the form of a matrix.

  The number of rows in this matrix multiplied by the number of columns gives us the total number of pixels in the image. An image with 20 columns and 20 rows will have dimensions of 2Ox2O=400.as displayed in figure 1.4

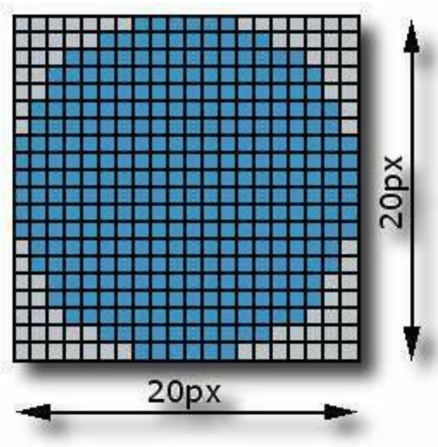

FIG. 1.4: 20x20 size image.

- **Resolution :**

  It is the pixel density of the image , resolution is usually measured in dots per inch (dpi) or pixels per inch (ppi) so 1 inch = 2.54 cm. The higher the resolution, the more detail the image can display, and it will appear clearer and more detailed.The high or low quality of an image depends on the relationship between pixel density and image size. Resolution : It is the pixel density of the image, resolutionis usually measured in dots per inch (dpi) or pixels per inch (ppi) so 1 inch = 2.54 cm. The higher the resolution, the more detail the image can display, and it will appear clearer and more detailed. The high or low quality of an image depends on the relationship between pixel density and image size.[3]

  as displayed in figure 1.5

FIG. 1.5: Example illustrating different image resolutions .

- **Depth :**

  The number of bits used to store the color of each pixel, which defines the range of colors available in the image. The greater the depth, the more accurate the image is in terms of colors and tonal gradations.

- **Noise :** This is the blurring or irregularity of the pixels in the image. Noise may be due to shooting conditions, camera settings, image pressure, or other factors.as displayed in figure 1.6

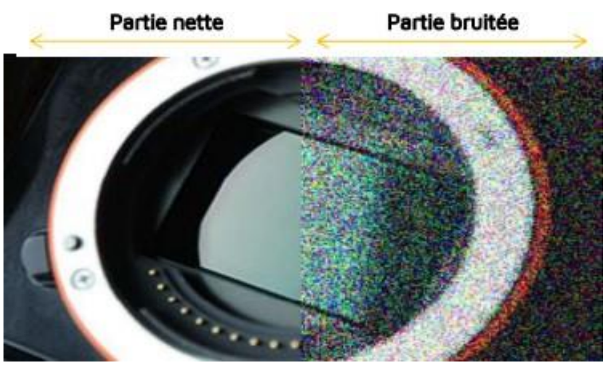

FIG. 1.6: Example of an image affected by noise .

- **Luminance :**

  The amount of light reflected by each pixel in the image determines the overall brightness of the image. It is also defined as the quotient of the luminous intensity of a surface by the apparent area of that surface.[4]

- **Contrast :**

  The difference between the highest and lowest luminosity values in an image. High contrast means that the clear parts of the image are too bright, and the dark parts are too dark. Which can be modified to improve readability.

- **Photometry :**

  The quantitative measurement of light in an image, which can be used to determine properties such as optical density or relative brightness Photometry is important in the field of photography because it measures the amount of light and color in an image, which helps determine image quality. It is often used in conjunction with other metrics, such as resolution and contrast, to rate image quality.

1.5.4 Digital Image Interpretation

1.5.4.1 Digital Image Color Coding

Interpretation of digital images can vary depending on the type of image being considered. Here is a brief description of the interpretation of digital images according to their color format [5-6]

- **Black and white image :** A black-and-white image contains only binary values (0 for black and 1 for white.). Interpreting a black-and-white image can include shape recognition, edge detection, and image segmentation.as displayed in figure 1.7

FIG. 1.7: binary image.

- **Grayscale image :** A grayscale image contains levels of gray ranging from 0 (black) to 255 (white). This range of shades of gray makes it possible to represent more or less light areas in an image, which can be useful for detecting more subtle features and shapes. It allows to get more nuances than simple black and white.as displayed in figure1.8

FIG. 1.8: Grayscale image .

- **Color image :** Color image : A color image contains information about the color of each pixel in the image, usually represented by RGB (red, green, blue) values. Colors are obtained by mixing the three primary colors .as displayed in figure1.9A color image can be interpreted using techniques such as object recognition or image classification to identify the main features of the image, or by using image segmentation techniques to separate the different parts of the image according to their color.

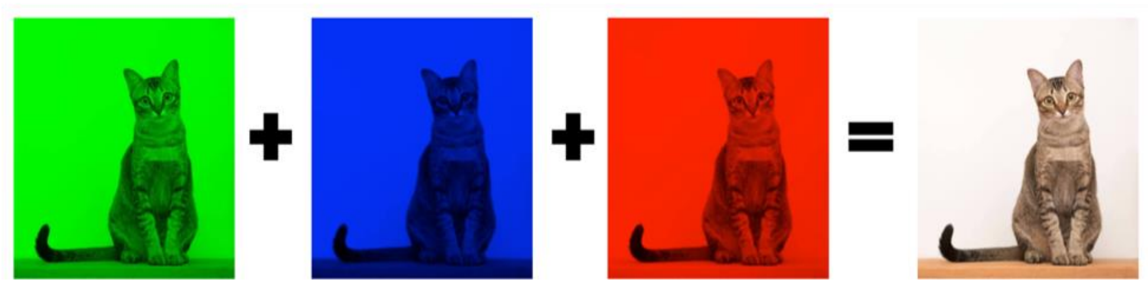

FIG. 1.9: color image.

1.5.4.2 Cumulative Histogram Of An Image :

A cumulative histogram of an image is a graph that shows the cumulative distribution of the pixel intensity values in the image. It is useful for visualizing the distribution of the pixel intensities and for identifying the most common intensity values in the image. [7]

- **Cumulative grayscale histogram of an image :** A cumulative grayscale histogram of an image is a graph that shows the cumulative distribution of the pixel intensity values in a grayscale image. The x-axis represents the possible pixel intensity values (from 0 to 255), and the y-axis represents the cumulative frequency of pixels with intensity values less than or equal to the corresponding x-axis value. The resulting graph shows how many pixels have intensities below a certain value, and can help identify whether an image is dark or bright.

- **Color image :** A color image can also have a cumulative histogram, but it would require multiple histograms - one for each color channel (red, green, and blue). The resulting graph would show how many pixels have intensity values less than or equal to a certain value for each color channel. This can help identify whether an image is dominated by certain colors or has a balanced color distribution.

1.5.5 Digital Image Processing

1.5.5.1 Definition Of Image Processing

Image processing is a field of computer science and mathematics that involves analyzing and processing digital images using various techniques and algorithms. It is used to enhance, modify and extract information from images, with the ultimate goal of improving image quality or extracting useful information from it.

Image processing technologies can be used in various fields, including medical imaging, remote sensing, surveillance, and digital imaging. The field of image processing includes a wide range of tasks, including image filtering, image recovery, image optimization, image segmentation, image recognition, and image compression.

1.5.5.2 Segmentation D'Image

Image segmentation is the image processing process that aims to divide an image into different regions or segments, based on certain properties such as color, texture, shape or brightness. The purpose of segmentation is to isolate the different parts of an image to facilitate processing, such as object recognition, image properties measurement, edge detection, and noise removal. as displayed in figure 1.10

There are various techniques for image segmentation, such as region-based segmentation, threshold segmentation, pooling segmentation, etc. Each of these technologies uses different standards and processing methods to divide the image into parts. Image segmentation is widely used in many fields, such as medicine, security monitoring, pattern recognition, computer vision, and image analysis. [8]

FIG. 1.10: Illustration of image segmentation (figure1: original image, figure2: segmented image) .

1.5.5.3 Filtering

Image segmentation is the image processing process that aims to divide an image into different regions or segments, based on certain properties such as color, texture, shape or brightness. The purpose of segmentation is to isolate the different parts of an image to facilitate processing, such as object recognition, image properties measurement, edge detection, and noise removal.as displayed in figure 1.11

There are various techniques for image segmentation, such as region-based segmentation, threshold segmentation, pooling segmentation, etc. Each of these technologies uses different standards and processing methods to divide the image into parts. Image segmentation is widely used in many fields, such as medicine, security monitoring, pattern recognition, computer vision, and image analysis.

FIG. 1.11: Example of a filter on an image.

1.6 Computer Vision Applications

Computer vision has a wide range of applications in various fields, including the following : [9]

- **Robotics :** Computer vision is used in robotics to help machines perceive and navigate their surroundings and make decisions based on the visual information they receive. It is used for many diverse tasks such as object recognition, tracking, movement planning, and obstacle avoidance.
- **Medical imaging :** Computer vision can be used to analyze medical images, such as X-rays, and MRI scans (magnetic resonance imaging). Which helps doctors and radiologists in diagnosis and treatment.
- **Security and Surveillance :**Computer vision is an important technology for self-driving vehicles because it helps them detect and identify objects in their environment, such as pedestrians, other vehicles, traffic signs, and traffic lights. This allows vehicles to make decisions and navigate safely.
- **Self-driving vehicles :** Computer vision is an important technology for self-driving vehicles because it helps them detect and identify objects in their environment, such as pedestrians, other vehicles, traffic signs, and traffic lights. This allows vehicles to make decisions and navigate safely.

- **Games :** Computer vision is used in games to enable games to interact with the real world. It is used for tasks such as motion tracking, gesture recognition, and facial expression analysis.

These are just a few examples of the many applications of computer vision. The technology has the potential to revolutionize various fields and industries by improving efficiency, accuracy, and safety and also its potential applications are expected to grow exponentially.

1.7 Conclusion

In this chapter, we have discussed one of the pillars of the broad field of artificial intelligence which is computer vision, and some definitions and concepts related to the latter that we will deal with and that will be very useful for the rest in Our application consists of recognizing and detecting the license plate in pictures and recognizing its constituent letters (numbers). We also talked about image processing and its basic concepts, In the next chapter, we will introduce the OpenCV library that helps us discover and identify panels

Chapitre 2

Presentation Of The Open Cv Libraries

2.1 Introduction

OpenCV (Open Source Computer Vision) is a popular open-source library for computer vision and image processing tasks. It provides a wide range of functions and algorithms that enable developers to perform tasks like image and video analysis, object detection and recognition, camera calibration, and more. OpenCV supports multiple programming languages, including C++, Python, and Java, and it is widely used in various fields such as robotics, augmented reality, and surveillance systems. In this chapter we present opencv, in the first place, we will talk about the history of the opencv library, then we will show what the opencv library is, after we talk about the opencv library, then we will show the form of the opencv library, after we talk about the uses of the opencv library ,Finally, we will conclude the chapter with a brief conclusion

2.2 History

OpenCV (Open Source Computer Vision Library) dates back to the late 1990s. It was originally developed by the Intel Corporation Research Center in Russia, as a project aimed at providing a common platform for researchers and developers working in the field of computer vision and making it accessible to a wider audience.

Here is a brief timeline of the history of OpenCV as displayed in figure 2.1 :

- **1999:** Gary Bradski and Adrian Kaehler started working on a library called CVML (Computer Vision Machine Learning) while at Intel Research. This library was later renamed OpenCV. Memechose pour le reste.
- **2000:** The first version of the library alpha OpenCV was launched as an open-source project, alpha versions are based on changing the color, blending and retaining part of the images using pixels.
- **2001 to 2005:** OpenCV Beta was released which focused primarily on photo or video background contrast and color arrangements.
- **2006:** The OpenCV project is taken over by Willow Garage, a robotics research lab in California. OpenCV 1.0 has been released which is built using the C language and adds a lot more Advanced image processing features.
- **2008:** OpenCV 1.1 released with support for C++, Python, and MATLAB interfaces, used for motion templates for videos Or photos and SIFT features and camera unification.
- **2009:** OpenCV 2.0 was released with a new C interface and improved support for parallel processing.
- **2015:** OpenCV 3.0 was released with improved support for deep learning and GPU acceleration.

- **2016:** Intel acquired the rights to OpenCV from Willow Garage and formed a new company called Itseez to maintain the library.
- **2018:** OpenCV 4.0 was released with a new deep learning DNN module, improved support for CUDA and OpenCL, and other improvements.
- **2021:** OpenCV 4.5.4 is the latest stable release of OpenCV as of early 2023.**[10]**

Today, OpenCV is widely used in many industries, such as robotics, autonomous vehicles, security, medical imaging, and more. It remains an active open-source project with a large community of contributors and users.

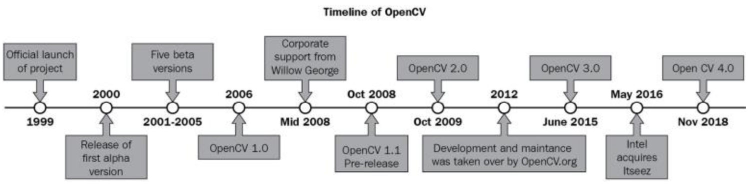

FIG. 2.1: timeline of opencv.

2.3 What Is OpenCV Library ?

OpenCV (Open Source Computer Vision Library) is a free, open-source library that includes programming functions mainly aimed at real-time computer vision applications. It provides a comprehensive set of image processing and computer vision functions, such as image and video I/O, image filtering, feature detection and extraction, object detection and tracking, camera calibration, machine learning, and more.

OpenCV is designed to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. It supports multiple programming languages, including C++, Python, Java, and MATLAB, and is highly portable across different platforms, including Windows, Linux, and Mac OS.

OpenCV is widely used in a variety of fields, including robotics, autonomous vehicles, surveillance, augmented reality, medical image analysis, and more. It is also popular among hobbyists and researchers who work on computer vision and machine learning projects.as displayed in figure 2.2

Some of the key features of OpenCV include :

- Support for multiple programming languages
- Comprehensive set of image processing and computer vision functions
- High performance for real-time applications
- Open-source and freely available to use
- Portability across different platforms

- Extensible and customizable with additional modules and plugins
- Integration with other popular libraries and frameworks, such as TensorFlow and PyTorch.

FIG. 2.2: Logo OpenCV.

The OpenCV library provides many very diversified functionalities allowing to creation of programs starting from raw data to go until the creation of basic graphic interfaces. It offers most of the standard low-level image and video processing operations.

2.3.1 Image Processing

It offers most of the classic low-level image processing operations :

- reading, writing, and displaying an image.
- Calculate the histogram of gray levels or color histograms.
- smoothing, filtering.
- image thresholding (Otsu method, adaptive thresholding).

•segmentation (related components, GrabCut).

- mathematical morphology

2.3.2 Video Processing

This library has established itself as a standard in the field of research because it offers a large number of state-of-the-art tools in computer vision such as :

- reading, writing, and displaying a video (from a file or a camera).
- detection of lines, segments, and circles by Hough Transform.
- face detection by the Viola and Jones method.
- cascade of boosted classifiers.
- motion detection, motion history.
- object tracking by mean-shift or Camshift.
- detection of points of interest.
- optical flow estimation (Lucas–Kanade method).
- Voronoi diagram.
- convex hull.
- Adjust an ellipse to a set of points by the method of least squares.

2.4 OpenCv Modules

OpenCV (Open Source Computer Vision) is a comprehensive library, offering a wide range of functions related to image and video processing, computer vision, and machine learning. It consists of several modules, each of which provides specific functions. Here are some of the main modules in OpenCV,as shown in figure 2.3 :

- **Core functionality :** This module provides the basic data structures, functions, and algorithms used throughout the library. Includes basic image and matrix processing functions, data structures, image and video I/O, memory management, and core functions used by all other modules.
- **Image processing (imgproc) :** This module includes image manipulation and processing functions. It provides features such as image filtering, color space transformations, edge detection, smoothing, morphological operations, histograms, etc.
- **Video analysis (video) :** This module provides functions and algorithms for vi- deo analysis and processing, including features such as video capture, background subtraction, and motion detection.
- **Détection d'objets (objdetect) :** ce module fournit des algorithmes pour détecter et reconnaître des objets dans des photos et des vidéos, tels que la détection de visage, la détection de piétons et l'extraction de caractéristiques HOG (Guided Gradient Graph).
- **3D vision(calib3d) :** This module provides basic geometry algorithms, stereoscopic correspondence algorithms, and 3D reconstruction functions, such as camera calibration, stereo calibration, and position estimation.

- **Video I/O :** Provides functions for reading and writing video files, capturing video streams from cameras, and accessing video codecs.
- **High level graphical user interface (highgui) :** This unit provides user interface functions for viewing and interacting with images and videos
- **2D Features Framework(features2d) :** This module contains feature detection and description algorithms such as SIFT and SURF.
- **Computational photography :** This module provides functions for creating special effects, HDR shooting, and other advanced image processing technologies.
- **Stitching :** This module provides functions for stitching images and creating panoramas.
- **Machine learning :** object recognition, face detection, and image segmentation.
- **Deep Neural Networks :** This module provides a deep neural network module that can be used for object detection, image classification, and more

Overall, OpenCV provides a comprehensive set of tools for working with images, videos, and other multimedia resources, making it an essential library for computer vision and machine learning applications.**[11-12]** .

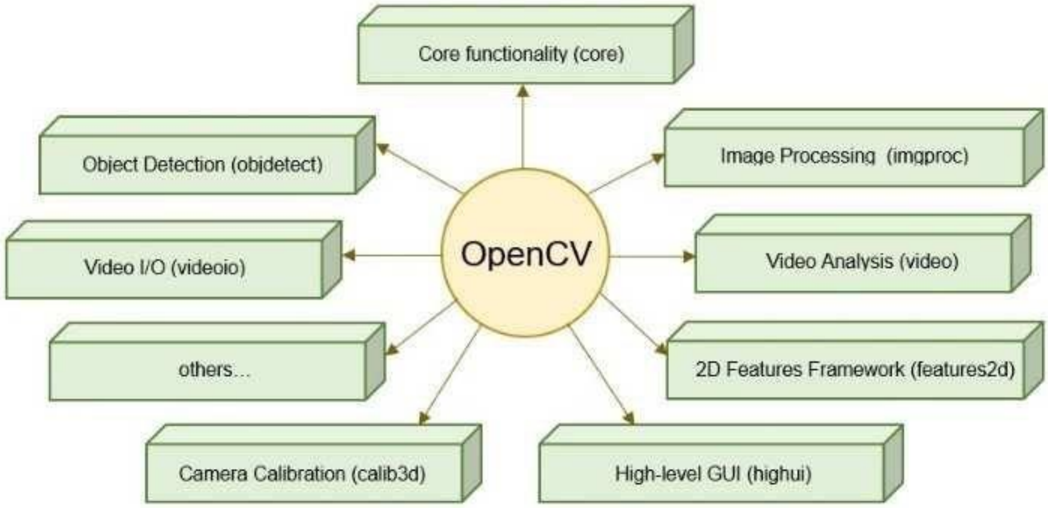

FIG. 2.3: Module of OpenCV.

2.5 Structure And Content Of Library OpenCv

OpenCV is a library of programming functions for computer vision and real-time image processing. It consists of five main modules, which themselves consist of several functions and algorithms. The following describes the main components of OpenCV and their associated algorithms, as shown in figure2.4 :

- **CV (Computer Vision) :** This component is dedicated to computer vision and contains algorithms for object detection, pattern recognition, segmentation, stereo-vision, 3D reconstruction, camera calibration, etc. Some of the most commonly used algorithms include :

  - SIFT (Scale-Invariant Feature Transform) : algorithm for detecting scale-invariant and rotation-invariant features.

  - SURF (Speeded-Up Robust Features) : robust and fast feature detection algorithm.

  - Hough Transform : algorithm for detecting lines, circles and other geometric shapes in an image.

- **ML (Machine Learning) :** This component is dedicated to machine learning and contains algorithms for image classification, segmentation, object detection, etc. Some of the most commonly used algorithms include :

  - SVM (Support Vector Machines) : algorithm for binary and multiclass classification.

  - K-NN (K-Nearest Neighbors) : algorithm for classifying and searching for near neighbors.

- **HIGHGUI :** This module provides functions for displaying images and videos in a window, as well as for handling mouse and keyboard events. Algorithms included in this module include :

  - The display of images and videos in a window (imshow)

  - The management of events related to the mouse and the keyboard (setMouseCallback, setKeyboardCallback).

- **CXCORE :** This module provides the basic data structures and functions needed for memory and error handling. Algorithms included in this module include :

  - Basic data structures (IplImage, CvMat)

  - Functions for memory management (cvCreateImage, cvReleaseImage)

  - Functions for error management (cvError, cvSetErrMode)

- **CXAUX :** This module provides additional functions for computer vision, such as memory management and multidimensional arrays. Algorithms included in this module include :

  - Memory management (malloc, free)

  - Management of multidimensional arrays (Mat, Scalar)

These examples are not exhaustive, but they give an idea of the algorithms and features available in each component of OpenCV. Each algorithm and feature has its own set- tings and configuration options to meet the specific needs of computer vision and image processing applications.**[12-13]**

.

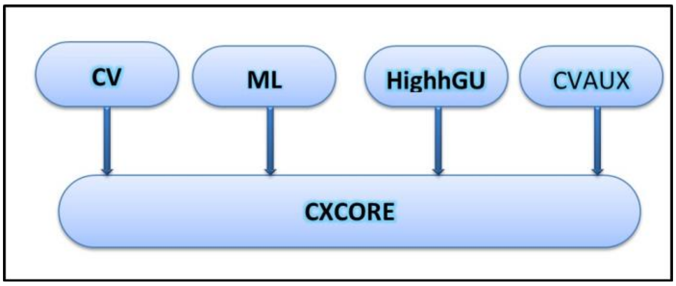

FIG. 2.4: Structure of OpenCV.

2.6 Using OpenCv

OpenCV can be used in many fields, such as robotics, industrial automation, video surveillance, augmented reality, face and object recognition, computer vision in general, and much more. Here are some examples of using OpenCV :

- **Object detection :** OpenCV can be used to detect objects in an image or video. This feature is useful in applications such as video surveillance, security, object recognition, and robotics.
- **Motion Tracking :** OpenCV allows motion tracking in video using optical flow, edge tracking, or motion detection algorithms. This feature is useful for video surveillance and gesture recognition.
- **3D reconstruction :** OpenCV can be used to reconstruct 3D scenes from multiple images using stereoscopic vision techniques. This feature is useful in augmented reality, 3D modeling, and mapping.
- **Image processing :** OpenCV provides many functions for image processing such as noise reduction, color correction, image format conversion, and more. This featureis useful for image editing, medical and scientific image processing, optical character recognition, and so on.
- **Machine learning :** OpenCV also provides machine learning functions, such as image classification, image segmentation, object detection, and so on. This feature is useful in the fields of artificial intelligence, natural language processing, speech recognition, gesture recognition, etc.

In short, OpenCV is a very versatile library that can be used for a wide variety of tasks and chosen for many computer vision and image processing projects, due to its ease ofuse and high performance.**[12-13]**

2.7 Conclusion

The OpenCV library is an indispensable tool in the field of computer vision and image and video processing. This library has changed many industries by enabling developers to perform complex tasks such as object detection, motion tracking, 3D reconstruction, image processing, and machine learning. The wide range of functions and algorithms available in the library, as well as its support for several programming languages, made it accessible and adaptable for diverse applications.

Chapitre 3

Conception Et Implémentation

3.1 Introduction

License plate detection and recognition is a technology that utilizes computer vision to detect and identify license plates from images of vehicles. In this project, we will be using the OpenCV library and C++ programming language to create a program that can detect and recognize license plates from images. The goal of this project is to develop a reliable and accurate system for license plate detection and recognition that can be used in various applications, such as traffic management, parking enforcement, and law enforcement, and we will provide an overview of the problem and the tools and techniques used, the implementation details, and the results of our experiments.

3.2 Problematic

The problems of this project revolve around the need for an effective and reliable system to detect and identify the license plate. and among these problematic aspects associated with this project are:

• Differences in license plate formats (different formats and designs across different regions and countries).
• Factors such as lighting, clogging and vehicle conditions.
• Pick up license plates under difficult conditions, such as low light, bad weather or blurry motion.
• Vehicle license plates reflect sunlight or artificial light sources that cause glare and reflections.
• Differences in angle and perspective lead to distortions that may affect the accuracy of the license plate.
• Real time and speed restrictions.

By overcoming these challenges, license plate detection and identification systems can provide valuable solutions in various areas, including traffic management, surveillance and law enforcement, contributing to increased security, and efficiency Transport systems.

3.3 working Environment

In this application we used the OpencCV library and tessract_OCR library and visual studio IDE, we present the hardware and software environments of our work.

3.3.1 Hardware Environment

An HP computer with the following specifications :

- **Processor :** Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
- **RAM :** 8.00 GB.
- **Hard disk :** 256 GB SSD.

3.3.2 Software Environment

In order to carry out our work, we used several development tools, in particular the OpenCV computer vision library and the C++ programming language, in this present chapter we will present the different libraries and modules that we used. in order to design a functional automatic license plate reading system that allowed us to save considerable time and thus focus on the most important part, which is the detection and reading of license plates.

- **Microsoft Visual Studio version 2022 :** Microsoft Visual Studio version 2022: This integrated development environment (IDE) provides a powerful platform for software development. It offers a user-friendly interface, code editing features, and debugging tools, making it an ideal choice for implementing your project.
- **OpenCV configuration in visual sudio version 2022:** In this part I provide information on how to configure OpenCV in Visual Studio, which is a popular integrated development environment (IDE) for Windows.

  To configure OpenCV in Visual Studio, follow these general steps :

  - Download OpenCV : Visit the official OpenCV website (https ://opencv.org) and download the appropriate pre-built OpenCV binaries for your operating system.
  - Install OpenCV : Extract the downloaded OpenCV package and follow the installation instructions provided by OpenCV. Be sure to note the installation guide.
  - First, we start by opening Visual Studio and creating an clr empty project. Start Visual Studio Click on Create a new project Select clr Empty Project and click on Next Specify Project Name and Project Location Click Create as displayed in figure(3.1)
  - We can now start adding new .cpp files or import existing code. Right-click on Source Files, then click on Add→New Item Click on C++ File (.cpp), specify Name and Location of the cpp file and click Add
  - The files which we have provided work for x64 configuration. Set Solution Platform as x64 Right-click on Project and click Properties

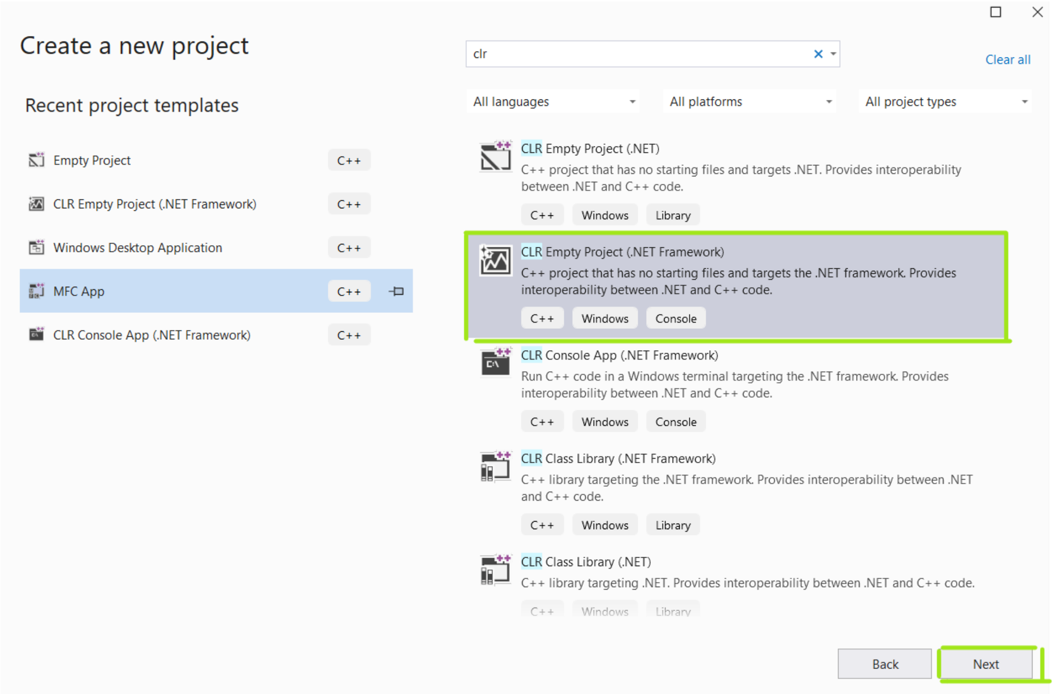

FIG. 3.1: Create a new project.

– We give paths to the OpenCV header files. This will, in turn, tell the compiler how the OpenCV library looks.

  Set All Configuration and x64 Platform Go to C/C++ → General → Additional Include Directories and Click Edit c :/ opencv/ build/ include Navigate and select the OpenCV include folder Click OK and then Click Apply as displayed in figure3.2

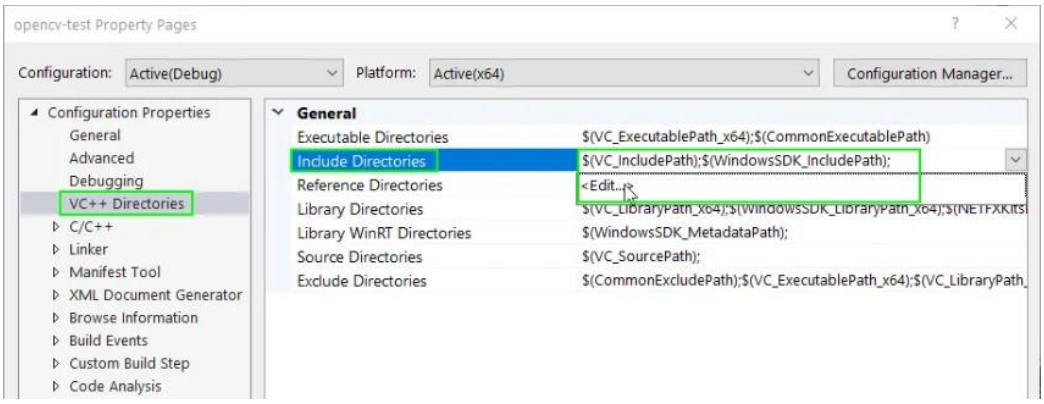

FIG. 3.2: Additional Include Directories.

- Now we give paths to the OpenCV library files. This will tell the linker where to get the functions or data structures of OpenCV when needed.

  Set All Configuration and x64 Platform Go to Linker → General → Additional Library Directories and Click Edit c :/ opencv / build /x64 /lib

Navigate and select the OpenCV lib folder Click OK and then Click Apply as displayed in figure3.3

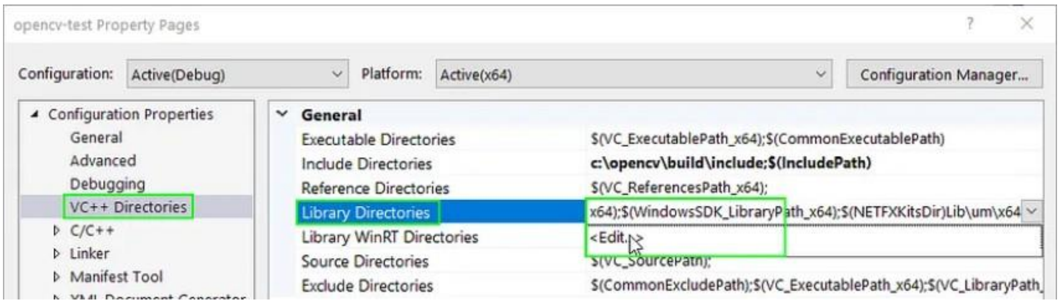

FIG. 3.3: Additional Library Directories.

- We need to list the .lib files for the modules which will be used.

  Set Configuration to Debug and x64 Platform Go to Linker → Input → Additional Dependencies and Click Edit Specify the list of all lib file for Debug Mode as displayed in figure3.4 , 3.5

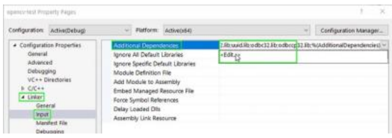

FIG. 3.4: Add Dependencies.

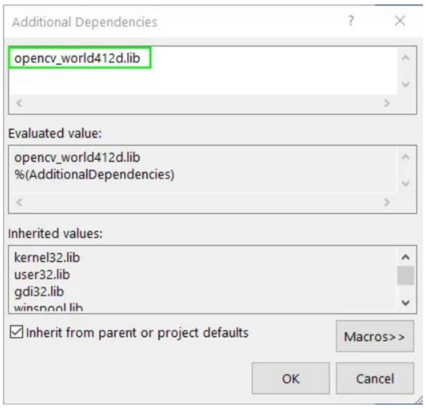

FIG. 3.5: Add opencv_world412d.lib.

These steps provide an overview of configuring OpenCV in Visual Studio. Specific details may vary depending on the version of Visual Studio and the version of OpenCV. It is recommended to refer to the official OpenCV documentation and Visual Studio documentation for detailed and up-to-date instructions.

- **Tesseract OCR :** Tesseract OCR is a widely used OCR engine. It is capable of Optical Character Recognition (OCR) to extract text from an image of a license plate, which can be useful in identifying alphanumeric characters on number plates. The Tesseract OCR library allows us to invoke the Tesseract executable file (tesseract.exe) in our project to extract textual information from detected number plates.

By setting up the environment with Microsoft Visual Studio version 2022 and integrating the OpenCV and Tesseract OCR libraries, we will have a comprehensive toolkit to implement a number plate recognition and detection system. Microsoft Visual Studio will provide the development environment, OpenCV will handle detection and identification of license plates, and Tesseract OCR will be responsible for extracting text from detected number plates.

This working environment will enable us to develop a robust and efficient number plate recognition and detection system, allowing us to accurately identify and manipulate number plates in images..

3.4 Steps To Detect And Recognize License Plate

License plate detection and recognition by the camera involves a series of steps aimed at identifying and recognizing alphanumeric characters on a vehicle's license plate. These steps can be summarized as follows :

3.4.1 Step 1 : Import Libraries

Start by importing the libraries you have installed in the environment. Importing libraries allows you to call and use their functions in the project.

```
#include <fstream>
#include <opencv2/opencv.hpp>
#include <iostream>
#include <string>
#include <tchar.h>
```

The following code includes two header files from the OpenCV library **(<opencv2/opencv.hpp> and <iostream>)**. Here is an explanation of what this code does :

#include <opencv2/opencv.hpp> : This line includes the main header file of the OpenCV library, which provides various functions and classes for computer vision andimage processing tasks.

#include <iostream> : This line includes the header file for C++ I/O stream operations. It is used here to enable the use of standard I/O streams such as std ::cout to display output messages.

#include <string> :This line includes the <string> header, which provides various string manipulation functions and classes. It allows you to work with strings in C++.

#include <tchar.h> :The line include <tchar.h> includes the <tchar.h> header file, which is specific to Windows programming and provides support for Unicode and different character encodings

By including these header files, the code gains access to the functionality provided by OpenCV and enables the use of standard I/O operations.

3.4.2 Step 2: Open Camra

We need to configure the camera device and access it in its code to capture the live video stream. Here is an explanation of how to achieve this.

The Code :

```
VideoCapture cap(0) ;
Mat img ;
```

The VideoCapture cap(0) line of code creates a video capture object from the device's default webcam or camera. Here is an explanation of this line of code :

VideoCapture : It is a class in the OpenCV library that allows capturing video streams from different sources, such as video files or cameras.

cap :This is the name of the variable that represents the video capture object. You can name it differently if you want.

(0) :This is the argument passed to the constructor of the VideoCapture class. A value of 0 indicates that you want to capture video from the first available camera on your system. If you have multiple cameras connected, you can specify another number to select a different camera.

Using this cap object, you can then read the frames of the video using the **cap.read()** method in a loop to get the frames of each frame of the streaming video.

3.4.3 Step 3 - Starte To Detect License Plate :

At this step. We need to use the xml file in the opencv library to identify and detect vehicle number plates through the camera, which is called **"harcascaderussianplatenumber"**, which is a specially trained file to detect License plate numbers. Haar sequence classifiers are algorithms based on object detection based on evaluating rectangular regions of an image for certain traits, in this case the classifier is wasteful using a dataset of images containing license plates, and taught to distinguish between positive samples (images with license plates) and negative samples (Pictures without license plates). The resulting classifier can then be applied to new images or video frames to detect license plates by scanning the image and selecting areas that match the properties of the license plates. **"haarcascaderussianplate number.xml"** contains the learned parameters of the trained classifier.

The Code :

```
CascadeClassifier plateCascade ;
plateCascade.load("haarcascade_russian_plate_number.xml") ;
if (plateCascade.empty()) {
cout << "xml file not loaded" << endl ;
}
```

plateCascade.load("haarcascaderussianplatenumber.xml") ; :This line of code loads an XML file containing the information needed by the Haar cascade classifier for license plate recognition

The Code :

```
vector<Rect> plates ;
pictureBox1->Size =
↪  System ::Drawing ::Size(cap.get(CAP_PROP_FRAME_WIDTH),
↪  cap.get(CAP_PROP_FRAME_HEIGHT)) ;
while (true)
{
cap.read(img) ;
plateCascade.detectMultiScale(img, plates, 1.1, 3) ;
for (int i = 0 ; i < plates.size() ; i++)
{
Mat imgCrop = img(plates[i]) ;

pictureBox2->Image = gcnew System ::Drawing ::Bitmap(imgCrop.cols,
↪  imgCrop.rows, imgCrop.step,
Imaging ::PixelFormat ::Format24bppRgb, (IntPtr)imgCrop.data) ;
imwrite("C :\\TempFile\\img_plate_number.png", imgCrop) ;
rectangle(img, plates[i].tl(), plates[i].br(), Scalar(25, 0, 255), 3) ;
```

vector<Rect> plates ; : This is a vector that will be used to store the delimitation rectangles of the license plates detected in the image.

plateCascade.detectMultiScale(img, plates, 1.1, 10) ; : This function uses the loaded cascade classifier (plateCascade) to detect license plates in the image (img). Parameters 1.1 and 10 respectively define the scale factor and the minimum number of neighbors required to validate the detection.

rectangle(img,plates[i].tl(),plattes[i].br(),scaler(25,0,255), 3) : this function used to draw a rectangle on an image. It takes as input several parameters, including the image on which the rectangle will be drawn, the coordinates of the upper left corner of the rectangle, the coordinates of the lower right corner of the rectangle, the color of the rectangle and the thickness of the line.

3.4.4 Step 4 - Recognition Number Plates :

At this stap, we need to use the tesseract_OCR library to recognize the text from a saved license plate image that has been detected

The Code :

```
string cmd_tesseract = "\"C :\\Program
↪ Files\\Tesseract-OCR\\tesseract.exe\"
↪ C :\\TempFile\\img_plate_number.png c :\\tempFile\\output -l eng
↪ --oem 1 --psm 13" ;
system(cmd_tesseract.c_str()) ;
ifstream infile("C :\\TempFile\\output.txt") ;
if (infile.is_open()) {
string ocrResult ;
getline(infile, ocrResult) ;
System ::String^ ocrResultText = gcnew
↪ System ::String(ocrResult.c_str()) ;
listBox1->Items->Add(ocrResultText) ;
infile.close() ;
}
plates.clear() ;
}
```

system(cmd_tesseract.c_str()) ; : This line executes the command stored in cmd_tesseract using the system() function. It runs Tesseract OCR on the specified input image with the provided options.

ifstream infile("C ://TempFile//output.txt") ; : This line opens the output file (output.txt) generated by Tesseract OCR for reading using an ifstream object named infile.

getline(infile, ocrResult) ; : This line reads a line from the opened file (output.txt) and stores it in the ocrResult variable. This line assumes that the file contains only one line of recognized text.

3.5 Implementation

This application is an application written in C++ language that allows detection and identification of vehicle number plates The application allows :

- Open the camera inside it to take photos or videos of cars.
- Uses the camera to detect and identify visible vehicle number plates.
- The application cuts the picture of the painting from the original picture taken.
- The application recognizes the numbers inside the number plate
- The application extracts the recognized numbers and saves them as text in the form of a file.

FIG. 3.6: application interface

The following diagram shows how the application works, with pictures showing each stage as displayed in figure3.8

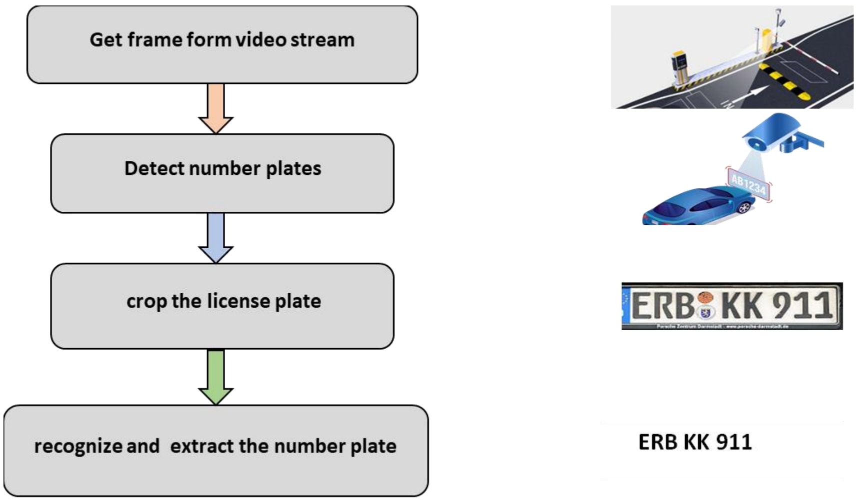

FIG. 3.7: How the app works.

3.6 Work Result:

In the work results of license plate detection and recognition, the system successfully determines the presence and location of license plates in the video streams through the camera. Moreover, the License Plate Recognition component shows great performance in extracting alphanumeric characters from detected plates, enabling conversion of license plate information into a machine-readable format. The system proves its ability to simplify data entry processes and reduce manual effort and errors. Overall, the results of the work on license plate detection and recognition demonstrate the development and successful implementation of an advanced system that improves efficiency, accuracy, and operational effectiveness in various areas that depend on license plate information.as displayed in figure3.8

FIG. 3.8: application interface

3.7 Conclusion :

In this chapter, we have designed and presented the practical side of our application, and we first talked about the problems of this project, and then the tools needed for our work environment to realize our application. Then we showed the different steps to configure visual studio and we also talked about the step needed to about detection and recognition license plates with screenshots of each functional part of our application with a description.

General conclusion

This project aimed to develop a license plate detection and recognition system using OpenCV and C++. Through the various chapters, we have explored the field of compu- ter vision, understanding its definition, purpose, and benefits. We also studied concepts related to images, such as their representation, characteristics, interpretation, and digital processing.

In the second chapter, we presented the OpenCV libraries in detail, exploring their history, structure, and content. We also discussed the practical use of OpenCV in the development of our license plate detection and recognition system.

The third chapter focused on the design and implementation of the system. We have described the work environment, both hardware and software. In addition, we presented the different steps required for the detection and recognition of license plates, detailing the actions required, such as importing libraries, opening the camera, detecting license plate registration, and plate number recognition.

By implementing our system, we were able to obtain satisfactory results. However, we also identified some issues, such as sensitivity to varying lighting and viewing angles. These challenges can be considered as avenues for improvement for future work. Future work could focus on developing advanced algorithms, integrating machine learning techniques, and refining the system's performance in challenging scenarios.

In conclusion, this project made it possible to develop a functional system for the detection and recognition of license plates using OpenCV and C++. This system has significant advantages in areas such as surveillance, traffic management, and road safety. However, there are still opportunities for improvement to refine the accuracy and robustness of the system. This work opens the way to new research perspectives in the field of computer vision and pattern recognition. The knowledge gained from this project can serve as a foundation for future research and development in the field of computer vision and its applications in license plate recognition.

Bibliography or reference

[1] "Advantages and Disadvantages of Computer Vision," Master in Artificial Intelligence and Deep Learning: University of Alcala - Madrid, Jan. 28, 2019. https://master-artificialintelligence.com/advantages-disadvantages-computer-vision (accessed May 14, 2023).

[2] "Base du dessin vectoriel -Illustrator- Pixellisation-Vectorisation," Imedias. https ://www.imedias.pro/cours-en-ligne/logiciels/illustrator/bases-dessin-vectoriel/ (accessed Mar. 26, 2023).

[3] C. Panda, « Qu'est-ce que la résolution d'image ? Comprendre les mégapixels, DPIetPPI»,ClippingPanda,21novembre2020.https://clippingpanda.com/fr/understanding-image-resolution/ (consulté le 28 mars 2023)

[4] Zeroual Djazia, "Implémentation D'un Environnement Parallèle Pour La Compression D'Image A L'Aide Des Fractales", Université de Batna, 2006.

[5] D.A.P. 24 octobre 2019,« IBM Developer », IBM Developer. http ://developer.ibm.com/articles/learn-the-basics-of-computer-vision-and-object-detection/developer.ibm.com/articles/learn-the-basics-of-computer-vision-and- object-detection (consulté le 28 mars 2023).

[6] « Image Processing :Techniques, Types, Applications [2023] ». https ://www.v7labs.com/blog/image-processing-guide, (consulté le 28 mars 2023).

[7] Kenneth Dawson-Howe. Histograms .In: A practical introduction to computer vision with OpenCV.2014: First Edition.35-47page.

[8] T. Mrinal, "Image Segmentation Explained | Built In," Mar. 13, 2023. https://builtin.com/machine-learning/image-segmentation (accessed Apr. 23, 2023).

[9] "What Is Computer Vision: Applications, Benefits and How to Learn It," Simplilearn.com, Apr. 28, 2021. https://www.simplilearn.com/computer-vision-article (accessed mars. 23, 2023).

[10]EDUCBA.[enligne].(lastupdated,7th,juner,2020).availableon:https ://www.educba.com/opencv-version/ ?source=leftnav.

[11]OpenCV. [en ligne]. (Generated on Thu Dec 29 2022 23:25:45 for OpenCV by doxygen 1.8.13) available on : https ://docs.opencv.org/4.x/d1/dfb/intro.html

[12]Adrian Kaehler and Gary Bradski. Learning OpenCV 3.Computer Vision in C++ with the OpenCV Library.December 2016:First Edition.

[13]Gary Bradski and Adrian Kaehler.Learning OpenCV.September 2008: First Edition.