REPUBLIC OF BENIN
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITY OF ABOMEY-CALAVI

INSTITUTE OF TRAINING AND RESEARCH IN COMPUTER SCIENCE

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj

THESIS

for obtaining a

Bachelor degree in computer science

Option : Internet and Multimedia

Presented by :
Gédéon Prince Yiségnon **GUEDJE**

NextVision: An AI-Powered Intelligent Video Surveillance Platform for Enhanced Security

Under the supervision of :
Dr. Eng. Vinasetan Ratheil HOUNDJI
Lecturer at IFRI, UAC

Maurice COMLAN	Dr (MA)	ENEAM,UAC	President
Eric ADJE	Eng.	IFRI,UAC	Examiner
Vinasetan Ratheil HOUNDJI	Dr (MA)	IFRI,UAC	Reporter

Academic Year **2022 2023**

Overview

Dedication

This bachelor's thesis is dedicated to my dear parents, whose unwavering support and constant encouragement have been my guiding light throughout my academic journey. Their love and faith in me have been instrumental in my success, and I am deeply grateful to them for their sacrifices and unwavering belief.

Acknowledgments

This project would not have been possible without the help of many people. I would like to express my sincere gratitude to my parents for their unwavering support and encouragement, which has been essential in my academic journey. I would like to express my most sincere thanks to Prof. **Eugène C. EZIN**, who is a source of motivation and inspiration for me, throughout my undergraduate studies, for his invaluable advice, his guidance expert and his constant support. I would like to acknowledge Dr. Ing. **Vinasetan Ratheil HOUNDJI**, my thesis supervisor, who agreed to supervise this work and guided me throughout the process. You will find here the expression of my gratitude.

I am also grateful to the **MIFY** team for giving me the opportunity to complete my internship within their organization, which was an enriching experience. Their support and expertise were essential to the success of this project, and I am grateful to them for their contribution to my academic and professional development.

Résumé

La sécurité occupe une place centrale, que ce soit pour protéger les biens ou assurer la sécurité des personnes. Les systèmes de surveillance traditionnels, bien qu'initialement conçus pour répondre à ces besoins, présentent des faiblesses notables. Les agents de sécurité, devant surveiller en permanence les écrans, peuvent parfois rater des séquences cruciales, mettant en péril la sécurité. De plus, déterminer le moment ou enregistrer les flux des caméras de surveillance peut s'avérer complexe voire fastidieux, et parfois rende ces séquences enregistrées inexploitables. C'est dans ce contexte que naît "NextVision", objet de notre mémoire. NextVision s'attache à exploiter le potentiel de l'Intelligence Artificielle pour révolutionner la vidéosurveillance. Se positionnant comme une plateforme de vidéosurveillance intelligente et à l'interface conviviale, elle repose sur une fusion habile de la vision par ordinateur et du traitement automatique du langage naturel pour la détection automatique des objets d'intérêt en temps réel. S'appuyant sur une architecture backend basée sur Django et un frontend dynamisé par Next JS, synchronisés en temps réel par des sockets, NextVision incarne une évolution continue, alimentée par l'apprentissage automatique, dans le but de renforcer la sécurité pour tous.

Mots clés : NextVision, Vidéosurveillance, Intelligence Artificielle, Vision par Ordinateur

Abstract

Nowadays, security is central, whether protecting property or ensuring individuals' safety. Although initially designed to address these needs, traditional surveillance systems exhibit notable shortcomings. Personal security, required to monitor screens constantly, may occasionally overlook crucial sequences, jeopardizing security. Determining when to record camera feeds can be intricate, and making these recorded sequences actionable presents an additional challenge. It is in this context that NextVision emerges. Our project endeavors to harness the potential of Artificial Intelligence (AI) to revolutionize video surveillance. Positioned as an intelligent and user-friendly video surveillance platform, it relies on a skillful fusion of computer vision and natural language processing for real-time automatic object detection. Supported by a Django-based backend architecture and a Next JS-powered frontend, synchronized in real-time via sockets, NextVision embodies a continuous evolution powered by machine learning to enhance security for all proactively.

Key words: NextVision, Video Surveillance, Artificial Intelligence, Computer Vision.

List of Figures

List of Tables

Acronyms and Abbreviations

AI : Artificial Intelligence viii
AWS : Amazon Web Services viii

CSS : Cascading Style Sheets viii
CV : Computer Vision viii

GPT : Generative Pre-trained Transformer viii

HTML : Hypertext Markup Language viii

IFRI : Institut de Formation et de Recherche en Informatique viii
IT : Information Technology viii

JSON : JavaScript Object Notation viii

LLM : Large Language Models viii

MOT : Multi-Object Tracking viii
MsAzure : Microsoft Azure viii
MVC : Model View Controller viii

NEXTVISION : Next Vision vi–viii, 15, 26–29, 39–43, 54, 55
NLP : Natural Language Processing viii, 2

OpenCV : Open Source Computer Vision Library viii

POS : Part Of Speech viii

TAL : Traitement Automatique de Langage viii

UAC : Université d'Abomey-Calavi viii
UML : Unified Modeling Language viii

XML : Extensible Markup Language viii

Glossary

API :

It is an Application Programming Interface. It is a standardized set of classes, methods, functions, and constants that serve as the facade through which software offers services to other software. viii

API Rest :

It is an Application Programming Interface (API or web API) that respects the constraints of the REST architectural style and allows interaction with RESTful (stateless) web services. viii

Asynchronous :

A programming paradigm enabling independent task execution without waiting for each other, particularly useful for handling time-consuming tasks and enhancing application performance and responsiveness. viii

Cache :

A technique of temporarily storing frequently accessed data in high-speed memory to reduce response time and improve application performance. viii

Celery :

An open-source distributed task queue system in Python, allowing asynchronous and scheduled execution of time-consuming tasks to improve application responsiveness. viii

Cloud :

The cloud refers to an infrastructure in which computing power and storage are managed by remote servers to which users connect via a secure Internet connection. Cloud services enable on-demand access to a variety of computing resources and services. viii

frameworks :

In computer programming, a framework, also called software infrastructure, is a coherent set of software components used to create the foundations and outline of the software. viii

Git :

Git is a decentralized version control software used for tracking changes in source code during software development. It allows developers to work collaboratively and maintain a history of their work. viii

JavaScript :

JavaScript is a scripting language primarily used in interactive web pages and is considered one of the core technologies of the World Wide Web. Along with HTML and CSS technologies, JavaScript enables the creation of dynamic and interactive web content. viii

library :

A library is a structured set that can be already compiled and ready to be used by programs to facilitate and allow the execution of certain processes. viii, 18

NextVision :

NextVision est un système de vidéosurveillance intelligent basé sur l'IA et la vision par ordinateur. Il détecte automatiquement les objets et actions en temps réel, optimise le stockage et permet une surveillance automatisée avec une interface conviviale. viii, 2

Redis :

An open-source, in-memory data structure store widely used as a caching solution to enhance the performance of web applications and systems. viii

Transfer Learning :

A machine learning technique that uses knowledge learned from one task or domain (pre-trained model) to improve performance on a different but related task. viii

WebSocket :

WebSocket is a communication protocol that enables real-time, bidirectional data transfer between a client and a server over a single, persistent connection. It allows for interactive and efficient communication in web applications, facilitating features like live updates, notifications, and chat functionalities. viii

YOLO :

YOLO (You Only Look Once) is an object detection system that can detect multiple objects within an image or video in real-time. It is known for its speed and accuracy, making it widely used in computer vision applications. viii

General Introduction

1. Context and justification

Video surveillance, as a modern surveillance technology, plays a vital role in the prevention and resolution of criminal and security incidents worldwide. CCTV[1] cameras have significantly reduced crime rates in many regions by deterring potential criminals and aiding in the identification and apprehension of suspects involved in illegal activities. Several research studies [5] have highlighted the positive impact of video surveillance on security: for example, in major cities across Africa where security challenges are prevalent, the implementation of surveillance systems has led to a noticeable decrease in thefts, assaults and acts of vandalism. In some cases, crime rates have dropped by more than 30 percent following the installation of surveillance cameras.

However, despite the obvious benefits of traditional video surveillance, there are still significant limitations. Conventional systems require constant monitoring by human operators, who can be prone to potential errors due to fatigue or lapses in attention. In addition, these cameras often record unnecessary video sequences, resulting in inefficient use of human and material resources. In addition, the recorded video is not always easily interpretable to extract valuable information, limiting its potential for automated decision-making. To address these issues, the NextVision initiative presents an Artificial Intelligence AI-based approach to video surveillance. By fusing computer vision and natural language processing, the platform automatically detects suspicious objects and behaviors in real-time video streams. This intuitive capability allows operators to ask questions in natural language and receive answers in real time, enhancing their interaction with the surveillance system. It also minimizes resource wastage by selectively recording relevant video sequences and optimizing human and machine resources.

2. Problematic

Conventional video surveillance faces significant challenges, such as the need for constant human monitoring, wastage of resources, and the inability to make decisions from the videos. To address these challenges, NextVision aims to develop an intelligent video surveillance system using NLP, capable of automatically detecting suspicious objects in real-time and enabling intuitive interaction

[1] A CCTV system essentially consists of a set of strategically placed cameras, which capture and transmit images to a video management system, which records and displays these images.

with operators. The goal is to enhance security, optimize resource utilization, and improve overall surveillance efficiency.

3. Objectives

The general objective of NextVison is to make video surveillance more intelligent, flexible, and easy to use by overcoming its usual limitations and constraints.

More specifically, we would like to,

- Develop an intelligent video surveillance system using AI, capable of automatically detecting and analyzing suspicious objects and activities in real time.
- Integrate computer vision technologies, specifically YOLO v8, to enable accurate and efficient object detection from video streams.
- Implement a natural language processing (NLP) module to allow intuitive interaction between operators and the surveillance system, enabling real-time queries and responses.
- Reduce wastage of resources by recording and storing only relevant video sequences containing significant events, thus optimizing hardware and storage utilization.
- Provide a user-friendly interface for operators to efficiently interact with the system and access relevant information quickly.
- Enhance overall security and public safety by proactively preventing and addressing potential security threats through intelligent video surveillance and quick responses.

4. Structure of the document

The document is divided into three chapters, namely:

- **The review of literature**. we will start by defining the key concepts of our work, and then we will present the existing technologies in the fields of Computer Vision and natural Language Processing (NLP), as well as the existing solutions in the domain of video surveillance;
- **Design and technical choices**. In this chapter, we will present in detail the material and the methods used, as well as the technical choices made for the development of our solution;
- **Result and discussion**: This is our thesis's third and last chapter. We present our proposed solution and the results obtained. These results are followed by perspectives that can be explored in the future to achieve a more efficient tool.

Chapter 1

Literature review

Introduction

This chapter aims at one hand to give the definition of some concepts and notions necessary for the good comprehension of our work. On the other hand, we talk about existing solutions and then give our critical opinions on them.

1.1 Clarification of the main concepts

Explore the fundamentals of Artificial Intelligence, Machine Learning, and Computer Vision. These concepts are vital to understanding Next Vision, an ever-evolving intelligent video surveillance platform. Discover how these essential domains drive the advancements of Next Vision in the realm of intelligent video surveillance.

1.1.1 Artifical Intelligence

1.1.1.1 Definition

Any technology that makes it possible to solve complex problems that one would have thought reserved for human intelligence. In other words, AI is the attempt to imitate human intelligence using a robot or software to replace tasks that humans can perform, but also tasks that surpass[6].

1.1.1.2 Importance and benefit of artificial intelligence

Humans and machines are generating data faster than it is humanly possible to absorb it and interpret it to make complex decisions. Artificial intelligence is the basis of all computer learning and represents the future of complex decision-making processes. AI is necessary for its potential to change how we live, work, and play. It has been effectively used in business to automate human tasks, including customer service work, lead generation, fraud detection, and quality control. In several areas, AI can perform tasks much better than humans. Particularly when it comes to repetitive, detail-oriented tasks, such as analyzing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors. Because of

the massive data sets it can process, AI can also give enterprises insights into their operations they might not have been aware of.

AI's reach spans various domains, from language comprehension to autonomous decisions, exemplifying its power in solving intricate issues. A compelling aspect is Machine Learning (ML), which lets computers learn from data.

1.1.1.3 Machine Learning

Machine Learning is a subset of artificial intelligence that enables a machine or system to learn and improve automatically. Instead of explicit programming, machine learning uses algorithms to analyze vast amounts of data, derive insights from this information, and then make informed decisions.

Machine learning algorithms enhance performance over time as they are trained and exposed to more data. As a result, machine learning models are the outcome – what the program learns when it executes an algorithm on training data.

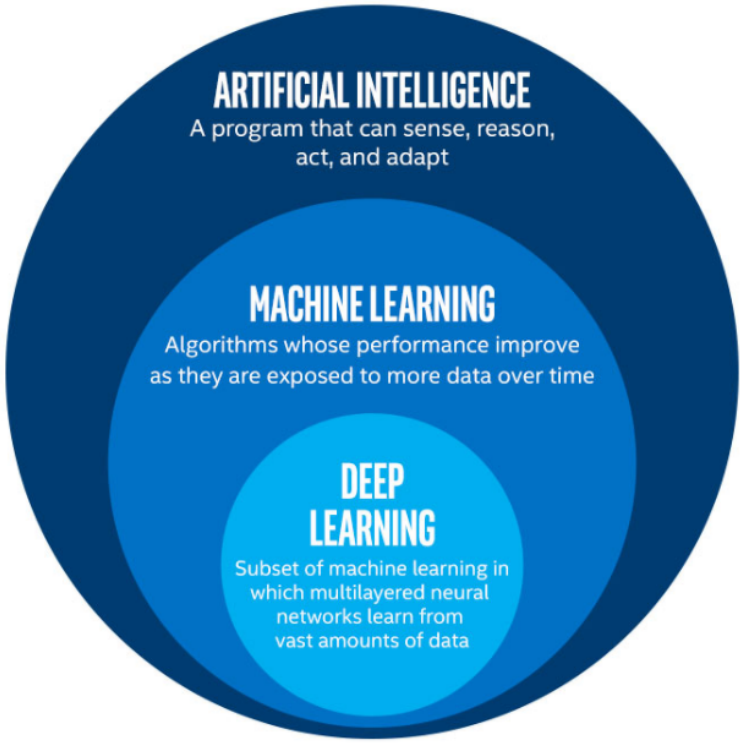

Figure 1.1: Artificial Intelligent -Machine Learning - Deep Learning[13]

1.1.1.4 Deep Learning

Deep Learning[10], a machine learning technique, draws its inspiration from the functioning of the human brain. Instead of just a simple model, it stacks dozens or even hundreds of layers of artificial neurons to create complex similar systems. These "layers" of neurons mimic how our brain processes information by receiving and interpreting data from previous layers.

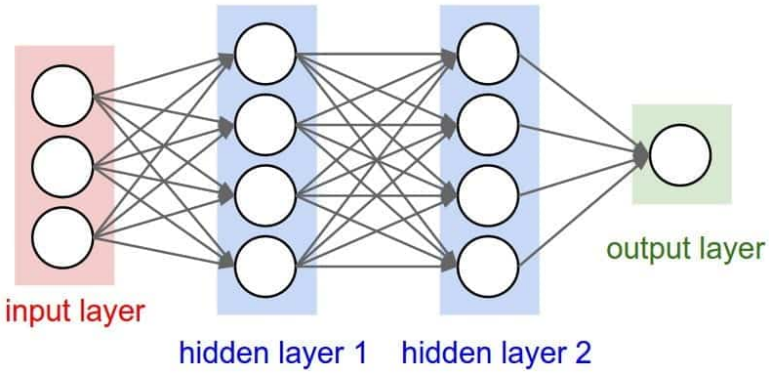

Figure 1.2: Artificial Neural Network[14]

1.1.2 Computer Vision

1.1.2.1 Definition

Computer vision [8] is an artificial intelligence technique that involves allowing computers to "see" and interpret images and video in the same way that police humans do. In other words, An interdisciplinary field that aims to enable computers to gain an understanding of what is being seen in images and videos.

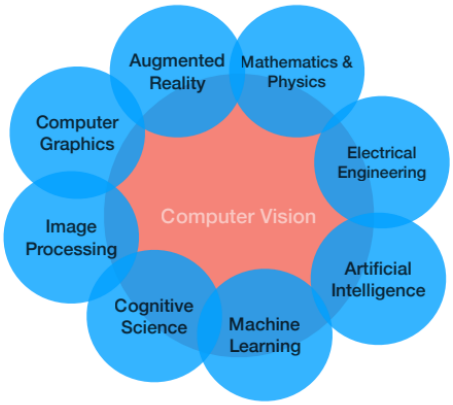

Figure 1.3: Fields Computer Vision[4]

1.1.2.2 General information on computer vision

- **What is an Image?**

  An image is a visual representation of something, while a digital image is a binary representation of visual data.

- **Understanding Image with Deep Learning**
  Most computer vision algorithms use something called a convolution neural network, or CNN. A CNN is a model used in machine learning to extract features, like texture and edges, from spatial data like images or videos.

  Like basic feedforward neural networks, CNNs learn from inputs, adjusting their parameters (weights and biases) to make an accurate prediction. However, what makes CNNs special is their ability to extract features from images.

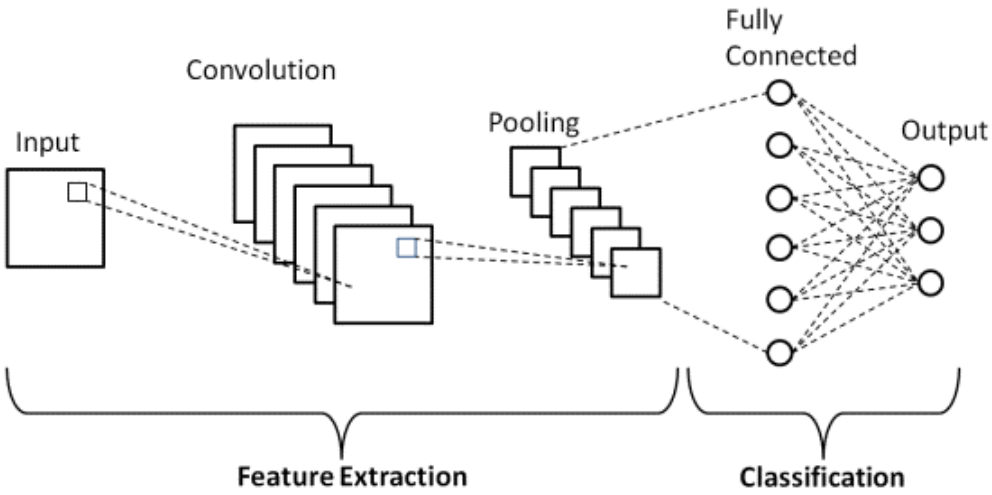

Figure 1.4: CNN Architecture[16]

- **Focus on convolutional neural network**
  CNNs [17] have achieved state-of-the-art performance on a wide range of image recognition tasks, including object classification, object detection, and image segmentation. They are widely used in computer vision, image processing, and other related fields, and have been applied to a wide range of applications, including self-driving cars, medical imaging, and security systems.

  Convolutional Neural Network Design :

  - The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order.
  - It is the sequential design that give permission to CNN to learn hierarchical attributes.
  - In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers.
  - The pre-processing needed in a ConvNet is kindred to that of the related pattern of neurons in the human brain and was motivated by the organization of the Visual Cortex.

1.1.2.3 Some major computer vision tasks

Object Detection
Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.[40] Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

The problem with this approach is that the objects of interest might have different spatial locations within the image and different aspect ratios. Hence, you would have to select a huge number of regions and this could computationally blow up. Therefore, algorithms like **R-CNN**,**YOLO** have been developed to find these occurrences and find them fast.

Object tracking

Object tracking is a computer vision application where a program detects objects and then tracks their movements in space or across different camera angles[34]. Object tracking can identify and follow multiple objects in an image. For example, a football recording studio could follow where a ball is in a photo.

Object tracking is a significant computer vision technology popular in augmented reality for estimating or predicting the positions and other applicable information of moving objects in real-time.

1.1.2.4 Object Detection Algorithms

Object detection finds and identifies things in images, and it's one of the biggest accomplishments of deep learning and image processing. We're going to explore the most popular algorithms while understanding their working theory, and benefits.

- **Region-based Convolutional Neural Networks** In the R-CNN[28] models, we try to extract the most essential features (usually around 2000 features) by making use of selective features. The process of selecting the most significant extractions can be computed with the help of a selective search algorithm that can achieve these more important regional proposals.
  Issues
  - Extremely slow
  - R-CNN model is not only the slow rate of training but also the high prediction time.
  - Bad candidate selections can occur at the initial step due to the lack of improvements that can be made in this particular step

- **Faster R-CNN**
  While the R-CNN[29] model was able to perform the computation of object detection and achieve desirable results, there were some major lackluster elements, especially the speed of the model.In the fast R-CNN method, the entire image is passed through the pre-trained Convolutional Neural Network instead of considering all the sub-segments. The region of interest (RoI) pooling is a special method that takes two inputs of the pre-trained model and selective search algorithm to provide a fully connected layer with an output
  Issues
  - One of the main limitations of the Faster R-CNN method is the amount of time delay in the proposition of different objects. Sometimes, the speed depends on the type of system being used.

- **Single Shot Detection**
  The single-shot detector[30] for multi-box predictions is one of the fastest ways to achieve the real-time computation of object detection tasks. SSD solves this issue by improving the frames per second to almost five times more than the Faster R-CNN model. It removes the use of the region proposal network and instead makes use of multi-scale features and default boxes.
  Issues
  - SSD architecture will typically perform worse than the Faster R-CNN for small-scale objects

- It suffers from decreasing the resolution of the images to a lower quality.

- **YOLO**
  YOLO[31] is one of the most popular model architectures and algorithms for object detection. This speed and accuracy is the main reason for its popularity.
  Issues
  - Failure to detect smaller objects in an image or video because of the lower recall rate
  - Many versions of YOLO : YOLOv3 , YOLOv5 , YOLOv6 , YOLOv8 , YOLOR, YOLOX

1.1.2.5 Object Tracking Algorithms

Object tracking aims at estimating bounding boxes and the identities of objects in videos. object tracking enables us to assign a unique ID to each tracked object, making it possible for us to count unique objects in a video.[38]

- **Simple Online And Realtime Tracking**
  Lean implementation of a tracking-by-detection framework.SORT[18] uses the position and size of the bounding boxes for both motion estimation and data association through frames. The IOU metric and the Hungarian algorithm are utilized for choosing the optimum box to pass on the identity.

- **DeepSort**
  DeepSort[19], an extension of SORT, is a CV tracking algorithm used to track objects by signing each tracked object with a unique ID. It introduces deep learning into the SORT algorithm by adding an appearance descriptor to reduce identity changes and thus make tracking more efficient.

- **ByteTrack**
  ByteTrack[20] is a multi-object tracking computer vision model. Using ByteTrack, you can allocate IDs for unique objects in a video for use in tracking objects. For example, you can track players on a football field and monitor them throughout a scene.
  BYTE is a simple and effective association method for MOT. This MOT technique is named BYTE as it considers each detection box as a basic unit of the tracklist like a byte in a computer program and the tracking method values every detection box and not just the ones with high scores

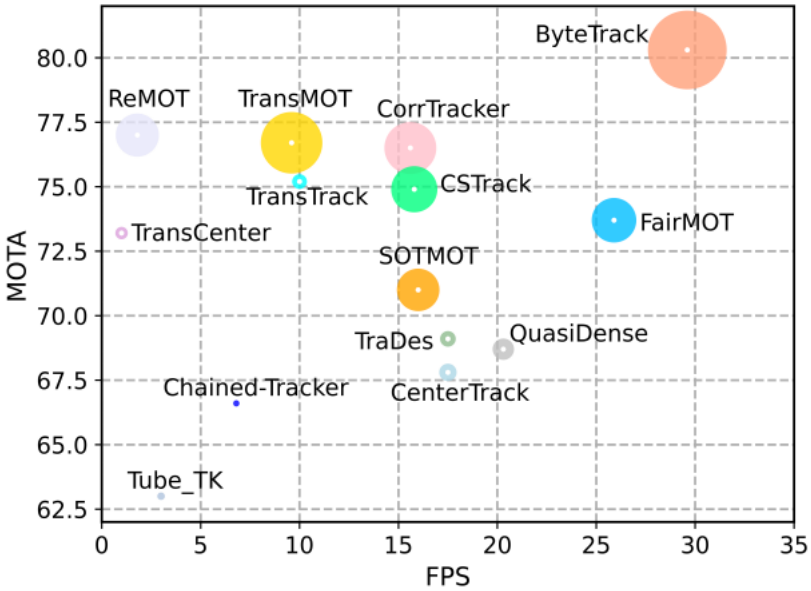

Figure 1.5: Algorithms Tracking Comparaison[46]

Description Metrics

MOT metrics are metrics used to evaluate the accuracy of tracking algorithms.

- **MOTA (Multiple Object Tracking Accuracy)** MOTA measures the accuracy of an MOT algorithm and considers three types of errors:
  - Missed detections: Objects present in the video but not detected.
  - False detections: Objects not present in the video but detected.
  - Identity switches: Incorrect assignment of identity to objects.

  MOTA is calculated as follows [21]:

$$MOTA = 1 - \frac{FN + FP + IDSW}{GT}$$

  Where:

  - FN is the number of missed detections.
  - FP is the number of false detections.
  - $IDSW$ is the number of identity switches.
  - GT is the number of ground truth objects.

  A higher MOTA score indicates better tracking accuracy, with a perfect score of 100 indicating no errors.

- **FPS (Frames Per Second)**[21]

  FPS measures the speed of an MOT algorithm, indicating the number of frames processed per second.A higher FPS score indicates a faster algorithm, which is crucial for real-time applications such as video surveillance.

1.1.3 Natural Language Processing

Natural language processing is the discipline of building machines that can manipulate human language or data that resembles human language in the way that it is written, spoken, and organized.

1.1.3.1 Transformers

The Transformer in NLP [41] is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease. It relies entirely on self-attention to compute representations of its input and output WITHOUT using sequence-aligned RNNs or convolution. NLP's Transformer is a new architecture that aims to solve tasks sequence-to-sequence while easily handling long-distance dependencies. Computing the input and output representations without using sequence-aligned RNNs or convolutions and relies entirely on self-attention.

1.1.3.2 Large Language Models

LLMs[22] are a type of AI that are currently trained on a massive trove of articles, Wikipedia entries, books, internet-based resources, and other input to produce human-like responses to natural language queries. That's an immense amount of data. LLMs work by taking sequences of text as input and generating context-based predictions. To do this, they use masks and tokens.LLMs use deep learning to analyze data and identify patterns and nuances of human language, including grammar, syntax, and context.

They typically rely on deep neural network transformer architectures, introduced by Google in 2017. These architectures enabled better context understanding and improved support for long sentences. Some examples of Large Language Models:

- **GPT**[23]: The Generative Pre-trained Transformer is one of the most well-known LLMs, notably used by ChatGPT and Microsoft Bing Chat. Its ability to generate highly coherent and contextually relevant text makes it suitable for various NLP tasks.
- **LaMDA**[24]: This is the initial LLM used by Google Bard, Google's AI chatbot.
- **BERT**[25]: The Bi-directional Encoder Representation from Transformers stands out from other LLMs due to its bidirectional features.
- **LLAMA**[26]:This is a collection of pretrained and fine-tuned large language models developed by Meta. The models in this collection range from 7 billion to 70 billion parameters. Specifically, the fine-tuned models, known as Llama 2-Chat, are optimized for dialogue use case.

1.1.3.3 ChatGPT, Artifical Intelligent Generative

ChatGPT[27] is a chatbot designed by the American company OpenAI, specializing in the field of artificial intelligence. Its main function: generate text to respond to Internet users' queries. The chatbot can generate text responses in several languages including French. It is also available in the form of an API. Its free version is based on natural language generation technology: GPT-3.5. This artificial intelligence model was trained to interpret text queries, also called prompts on ChatGPT, in order to generate a relevant response that respects the conditions indicated by the user.

1.2 Existing solutions

In this study, it's important to look at solutions that already exist, especially those using artificial intelligence and computer vision. These existing solutions help us understand the current technology landscape and can give us useful ideas for our own research. In this section, we'll take a closer look at some of these solutions, talking about what's good about them, where they fall short, and how they relate to our own work.

1.2.0.1 KiwiVision

KiwiVision Security Video Analytics[36] transform your video into smart, actionable information that helps you detect and understand emerging events so you can make the right decision at the right time.

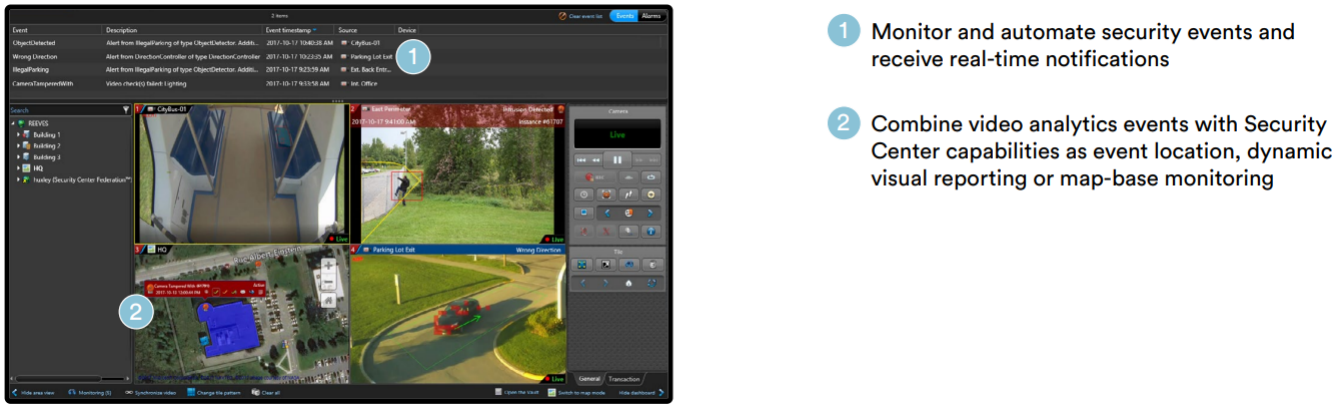

Figure 1.6: Interface KiwiVision Software[36]

1.2.0.2 Frigate NVR

Frigate[37] is an open-source NVR built around real-time AI object detection. All processing is performed locally on your own hardware, and your camera feeds never leave your home. It can detect and classify nearly a hundred different object types by using OpenCV and TensorFlow. Based on these detection events, Frigate generates short clips that can be easily browsed and filtered. This makes looking through footage much easier.

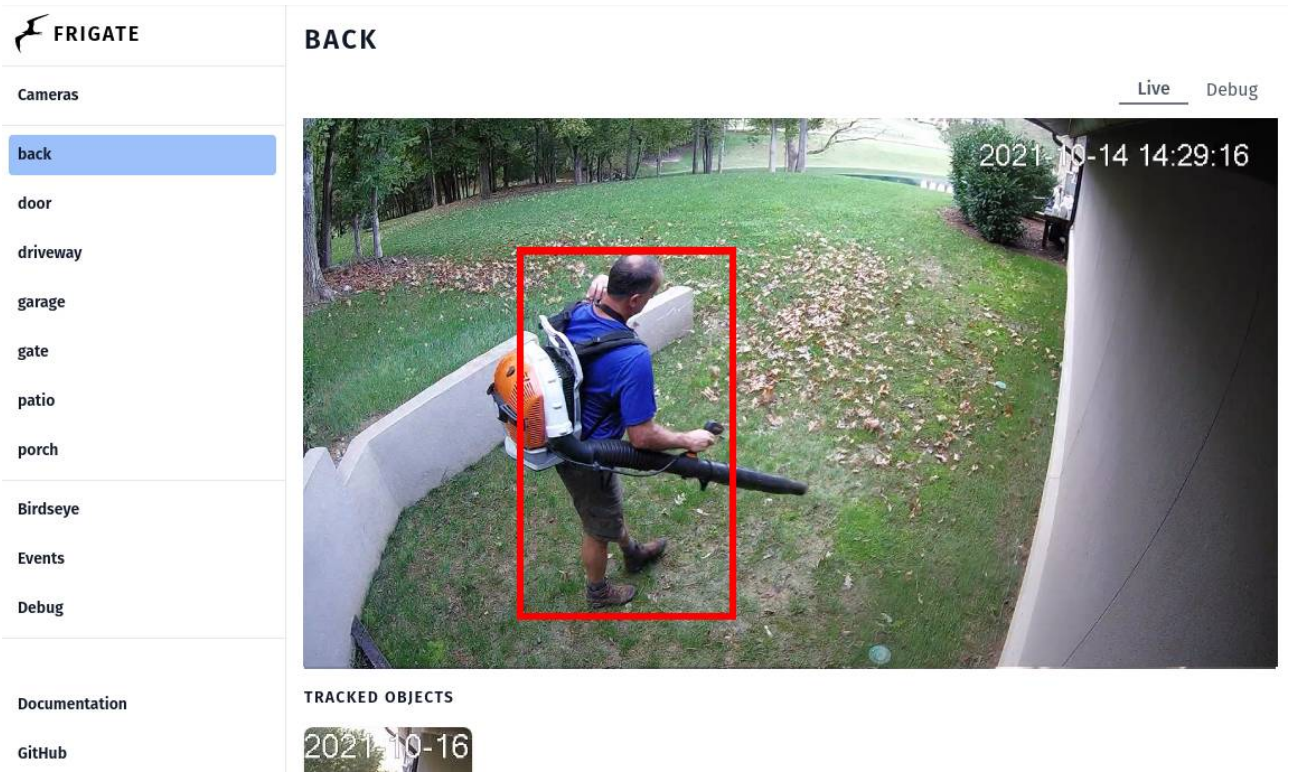

Figure 1.7: Interface Frigate NVR[37]

1.2.1 Amazon Rekognition

Amazon Rekognition[42] is a fully managed computer vision service that enables developers to analyze images and videos for a variety of use cases including identity verification, media intelligence, custom industrial automation, and workplace safety – no ML skills required. With Amazon Rekognition, you can identify objects, people, text, scenes, and activities in images and videos, as well as detect any inappropriate content. Amazon Rekognition also provides highly accurate facial analysis and facial search capabilities that you can use to detect, analyze, and compare faces for a wide variety of user verification, people counting, and public safety use cases.

1.2.2 Microsoft Video Indexer

Microsoft Video Indexer is a video analytics service that uses AI to extract actionable insights from stored videos.[47] It offers a visual demo and a programming interface (API) compatible with multiple languages such as Java, PHP, Python, and many others. This service allows for the extraction of relevant information from videos, including text and annotations. Once the video is uploaded, indexing starts automatically, requiring some waiting time until completion. This indexing provides text for context definition, enables sentiment analysis, and transcribes all the audio content of the video. Furthermore, the service is capable of tracking objects present in the video.

1.2.3 Google Cloud Video Intelligence

This cloud service allows an application to analyze the content of videos without having to develop anything. Google Cloud Video Intelligence offers multiple avenues for seamlessly integrating machine learning video intelligence models into your applications and websites through its API. With this service, you can harness the power of accurate video analysis, enabling the recognition of over

20,000 objects, places, and actions within videos. Additionally, it empowers you to extract rich metadata at the video, clip, or image level, providing comprehensive insights into your media content.[43]

1.3 Presentation of the limitations of existing solutions

Although the existing solutions mentioned above offer a range of functionalities, they often fall short in certain respects. Let's take a closer look at these limitations:

- **Lack of Security Specialization:** Generalist solutions such as Google Cloud Video Intelligence and Amazon Rekognition offer broad image and video analysis capabilities but may lack specialized features for nuanced security applications.
- **Technical Expertise Requirements:** Solutions like Frigate NVR require technical knowledge in Docker, posing a barrier for non-technical users.
- **API-Based Limitations:** The effectiveness of API-driven services like Amazon Rekognition and Google Cloud Video Intelligence depends on the user's ability to integrate them into existing systems, which can be challenging for those without programming skills.
- **Complex and Non-Ergonomic Interfaces:** The user interfaces of solutions like KiwiVision Security Video Analytics can be complex and non-intuitive, hindering efficient use and adoption.
- **Cost and Accessibility Issues:** Usage-based pricing models of services like Amazon Rekognition and Microsoft Video Indexer may become expensive and require a credit card, limiting accessibility.
- **Rigidity and Lack of Customization:** Many solutions, including Microsoft Video Indexer and Google Cloud Video Intelligence, offer limited flexibility and may not align with specific customer needs.

1.4 Our solution: NextVision

In response to identified limitations in existing solutions, we've developed **NextVision**, an innovative platform at the intersection of **AI**, **Computer Vision**, and **Natural Language Processing**. NextVision is divided into three key modules: **Vision**, which handles computer vision tasks like object detection and tracking; **Language**, which translates textual queries into system commands; and a real-time system based on **sockets** for instant communication.

Our containerized solution ensures **flexibility** and **scalability**, utilizing **sockets** and **asynchronous tasks** for a seamless experience. NextVision enables users to upload and stream video feeds, issue **natural language queries**, and program events, simplifying **intelligent video surveillance**

Figure 1.8: NEXTVISION Logo

1.5 Comparative table of solutions

The table below makes a comparison of the different solutions on the following criteria: entry, online accessibility, collaborative work, drawing automation, automatic source code generation and diagram export.

Table 1.1: Comparison of Existing Solutions vs. NextVision

Criteria	**Google Cloud Video API**	**Microsoft Video Indexer**	**Kiwivision**	**Frigate NVR**	**Amazon Rekognition**	**NextVision**
Object Detection	Yes	Yes	Yes	Yes	Yes	Yes
API	REST	REST	No	Yes	Yes	No (Socket-based)
Coding Required	Custom integration	Custom integration	No	Required	Custom integration	Minimal (Modular Design)
User-Friendly Interface	Moderately intuitive	Moderately intuitive	Moderately intuitive	Expertise needed	Non-tech friendly	Highly intuitive
Video Input Source	Video file, IP camera with config	Video File	Live Camera with config	Video File, IP camera with config	Video File	Live Cameras, Video Files (No extra config)
Price	Pay-as-you-go	Free trial or pay-as-you-go	Pay-as-you-go	Free	Pay-as-you-go	Flexible, based on modules used
Specialization	General AI	General AI	Security and Surveillance	Security and Surveillance	General AI	Security and Surveillance
Customization	Limited	Limited	Limited	Limited	Limited	High (Modular & Adaptable)
Owner	Google	Microsoft	Genetec	Open Source	Amazon	Ourself

Description

- **Object Detection:** Indicates whether the solution offers the capability to detect objects in videos. All listed solutions, including NextVision, support this functionality.
- **API:** Refers to whether the solution provides an Application Programming Interface (API) for developers to integrate it into other applications. Most solutions, except NextVision (which uses socket-based communication), offer API capabilities.
- **Coding Required:** This criterion describes the level of custom coding necessary to configure or customize the solution. While solutions like Google Cloud Video API and Microsoft Video Indexer require custom integration, NextVision requires minimal coding due to its modular design.
- **User-Friendly Interface:** Evaluates the intuitiveness and ease of use of the solution's interface, especially for non-technical users. NextVision is noted for its highly intuitive interface, contrasting with other solutions that may present complexities.
- **Video Input Source:** Indicates the types of video input sources supported by each solution,

such as video files, IP cameras, or live cameras. NextVision notably supports live cameras and video files without additional configuration.

- **Price:** Details the pricing model of each solution. Options range from pay-as-you-go models to free or flexible pricing based on modules used, as is the case with NextVision.
- **Specialization:** Describes the focus or specialization of each solution. NextVision is specifically tailored for security and surveillance, differentiating it from other more general AI solutions.
- **Customization:** Indicates the level of customization available for each solution. NextVision offers high adaptability with its modular and adaptable design. *
- **Owner:** Identifies the company or organization responsible for each solution. NextVision is our solution.

Conclusion

In this chapter, we introduce our solution for intelligent video surveillance. We assess existing solutions and identify the need for a more flexible and comprehensive approach. NextVision represents a paradigm shift, combining cutting-edge computer vision, natural language processing, and real-time communication technologies.

Our journey with NextVision began by analyzing the requirements and constraints of intelligent video surveillance. This informed our technical and organizational choices, which we'll detail in the following chapters. It is a dynamic web application, offering adaptability and responsiveness in security monitoring.

Chapter 2

Design and technical choices

Introduction

Our objective in this research project is to create an API that simplifies the intelligent management of surveillance cameras, making it user-friendly. This API incorporates a computer vision module capable of identifying specific elements in videos. Furthermore, it includes a text acquisition and data extraction module, enabling the understanding of user requests expressed in text and translating them into appropriate commands. In addition to this API, we are also developing an interactive web application that leverages this API. This chapter outlines the steps we have taken to develop our solution.

2.1 Development environment

2.1.1 Tools and Technologies

In our smart video surveillance project, we'll use Python[1] for its strong data handling and AI capabilities. We'll also use JavaScript[2] to create interactive user interfaces. For smooth user experiences and top-notch performance, we'll rely on Next.js for server-side rendering. Behind the scenes, Django, known for its reliability, will be our backend choice, organized using the mvc[3] approach for data management. Through Django's Channels extensions, we'll set up real-time communication to enhance our advanced security system.

2.1.1.1 Python

Python is an interpreted, cross-paradigm, cross-platform programming language. It promotes imperative, functional, and object-oriented programming[15]. It has dynamic typing, an automatic memory garbage collector, and an exception-handling system. It has an impressive collection of library that can be used for various tasks. Its major drawback is its relative slowness in the execution of the

[1] Python is an interpreted, cross-paradigm, and cross-platform programming language. It promotes structured, functional, and object-oriented imperative programming.
[2] JavaScript is a scripting language primarily used in interactive web pages and is an essential part of web applications.
[3] MVC: Model View Controller

code when the processing becomes complex. But, Python's strength lies in its flexibility, making it a preferred choice for diverse applications, including data science and artificial intelligence.

Its notable prominence in the fields of AI and data science is particularly significant. Python provides access to cutting-edge technologies like GPT APIs, effectively enhancing the capabilities of our intelligent video surveillance system.

- **OpenAI Python library** [58] provides convenient access to the OpenAI API from applications written in the Python language. It includes a pre-defined set of classes for API resources that initialize themselves dynamically from API responses which makes it compatible with a wide range of versions of the OpenAI API.
- **Open CV**[50] is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including object detection, face recognition, and tracking.
- **NumPy**[55] is a free and open-source software library for performing scientific calculations with Python. It introduced easier management of multidimensional arrays, derived objects (such as hidden arrays and matrices) as well as functions and for fast operations on these arrays (mathematics, logic, sorting, selection)[55].

2.1.1.2 Django

Django[51] is a high-level Python web framework encouraging rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django also has extra features like Channels, which helps with real-time communication, and REST framework, which makes it easy to create interfaces for other apps. These tools enhance our intelligent video surveillance system's capabilities and help it work smoothly with other technologies.. Its, as a modern framework, supports both WSGI and ASGI, allowing developers to choose the appropriate protocol based on their application's requirements.

2.1.1.3 NextJS

Next.js[52] is a React framework that allows you to build supercharged, SEO-friendly, and extremely user-facing static websites and web applications using the React framework. Next.js is known for the best developer experience when building production-ready applications with all the features you need.It has hybrid static and server rendering, TypeScript support, smart bundling, route prefetching, and more, with no extra configuration needed. It extends the original Facebook React library and the create-react-app package to provide an extensible, easy-to-use, and production-proof React framework.

2.1.1.4 Docker

Docker[53] is a tool for packaging an application and its dependencies in an isolated container, which can be run on any server. It is an open-source platform for developing, shipping, and running applications in containers. Containers are lightweight, portable, self-contained software units that bundle together an application and its dependencies. This ensures consistent development, testing, and

deployment of applications in different environments. Docker enables developers to bundle an application and its dependencies into a single container, which can then be easily deployed and run on any Docker-compatible platform. This eliminates the "it works on my machine" problem, as the containerized application will work in the same way regardless of the environment in which it is deployed. Docker also makes it easy to scale and manage containerized applications using a container orchestration platform such as Kubernetes. This automates the scaling, deployment, and management of containers across a cluster of machines. Docker also offers a centralized repository for storing and sharing container images, called Docker Hub, which makes it easy for developers to find and use pre-built images for their applications, or to share their own images with others.

2.1.1.5 LangChain

LangChain[57] is a framework designed to simplify the creation of applications using large language models (LLMs). As a language model integration framework, LangChain's use cases largely overlap with those of language models in general, including document analysis and summarization, chatbots, and code analysis. It's a library that allows us to build more advanced apps around LLMs like OpenAI's GPT-3 models or the open-source alternatives available via Hugging Face.

2.1.1.6 YOLO V8

According to Review Litterature on description algorithms Object Detection 1.1.2.4, YOLO is an object detection algorithm based on a Convolutional Neural Network that is capable of detecting objects in an image or video in a single step. This makes it significantly faster than traditional object detection algorithms like R-CNN, which require multiple steps to detect an object.

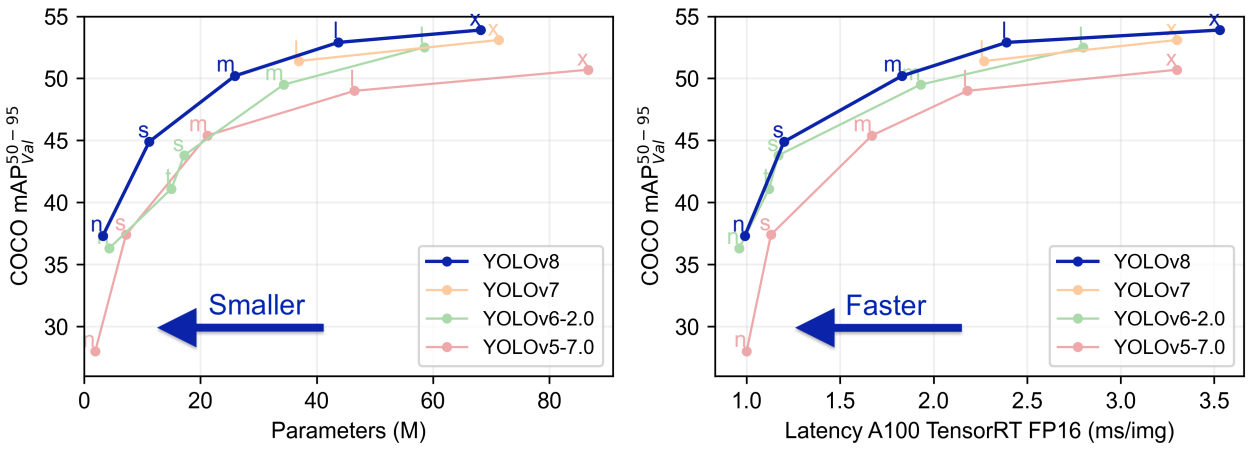

Figure 2.1: Comparaison YOLO versions[44]

The original YOLO[44] (You Only Look Once) was developed by Joseph Redmon using a custom framework called Darknet. Darknet is a versatile research framework written in low-level languages and has been responsible for creating a series of state-of-the-art real-time object detection models in computer vision, including YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, and YOLO-NAS.YOLOv8 represented the cutting-edge in object detection systems.

2.1.1.7 ByteTrack

ByteTrack[45] is a computer vision model designed for multi-object tracking. With ByteTrack, you have the capability to assign unique IDs to objects within a video, facilitating the tracking of these objects. ByteTrack is designed for real-time processing and offers faster inference speeds due to its lightweight backbone network. It is more suitable for applications where speed is a critical factor.

Justification

While we have chosen Django as our backend framework due to its maturity and extensive library support, we have also explored other options. Flask and FastAPI, for instance, were considered for their lightweight nature and suitability for medium-sized projects. In the realm of frontend frameworks, Next.js was selected for its user-friendliness, SEO advantages, and readiness for production. Nevertheless, React and Angular remain popular alternatives, each with its own set of benefits. Regarding containerization, Docker emerged as our top choice thanks to its widespread adoption, robust ecosystem, and comprehensive feature set. Nonetheless, Podman and Containerd are viable options. For object detection, YOLO V8 was chosen for its real-time performance, although alternatives like Faster R-CNN and SSD were contemplated to meet specific requirements for precision and speed. Each of these alternatives underwent thorough examination to ensure that our technological choices align seamlessly with our project's needs.

2.2 Our approach

Our approach focuses on a specific angle within the field of intelligent video surveillance. We highlight the detection of objects and the understanding of requests expressed in natural language. All of this takes place in the context of security, where technological solutions are gaining in importance to protect property and people. By using detection tools like YOLO (You Only Look Once), we create a basis for extracting relevant information from video streams. In addition, thanks to LLMS (Language and Multimodal Semantics), our system can understand and respond to questions formulated in everyday language. This methodical approach forms the solid foundation of our intelligent video surveillance system, enriching the dimension of security and visual analysis.

2.2.1 Overview

In the context of our thesis on AI-based video surveillance intelligence, we focus on creating a real-time system. Our goal is to develop a platform that not only monitors live video streams but also instantly analyzes data to detect objects of interest, events, or anomalies.

Our approach is built on three fundamental axes:

- Designing a Versatile Application: We are setting up a basic application capable of reading and streaming video feeds in real time, regardless of their source. This enables us to capture real-time insights and respond instantly to various situations.
- Training AI Models: We are developing and training deep learning models capable of rapidly detecting objects of interest in each frame of the real-time video feed. This allows us to identify and track crucial security elements in real time.

- Natural Language Processing Module: We are integrating a Natural Language-based module that enables users to formulate real-time text queries to interact with the system. This module instantaneously analyzes requests and translates them into commands to execute real-time actions.

This approach is founded on the collection and instant analysis of real-time video information, thereby providing an intelligent video surveillance solution that can react swiftly to critical events and allow real-time interaction with the system. This real-time data processing capability is a crucial element in enhancing both the security and performance of our system.

2.2.2 Object Detection Model Development

For the development of our object detection model, we opted for YOLO (You Only Look Once) version 8, which represents the cutting edge in terms of speed and accuracy. This choice was driven by its exceptional real-time capabilities compared to other models.

To ensure our model's accuracy and effectiveness, we carefully curated a dataset that encompasses our objects of interest.

Step 1 : **Creation of our customized dataset for security**

In the first step, we dedicated significant effort to creating a customized dataset tailored specifically for security applications. This dataset included a diverse range of images and videos featuring our objects of interest, which are 'Pistol,' 'Person,' 'Fire,' 'Cigarette,' and 'Phone.' These objects were carefully annotated to provide accurate labels for training. To streamline this process, we leveraged the robust annotation platform, Roboflow, which greatly simplified the annotation and organization of our data.

- **Roboflow Universe,The largest resource of computer vision datasets and pre-trained models.**

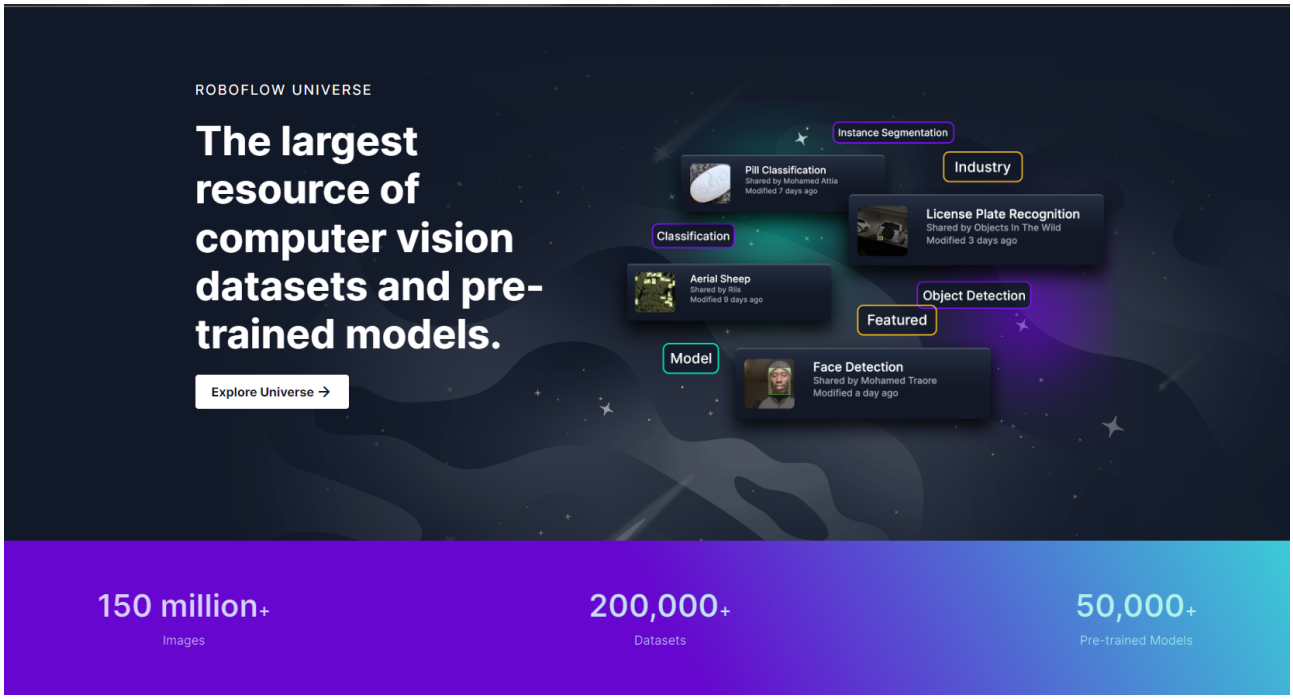

Figure 2.2: Roboflow Universe, resource for computer Vision[12]

To enhance the effectiveness of our security system, we developed a specific dataset to train YOLO models, a type of machine learning model for object detection. Our approach was to consult and merge various public datasets to include five target objects specific to our needs.

In addition, we integrated data from the COCO[54] (Common Objects in Context) dataset, which already included two of the objects we wanted to detect: people and knives. The aim of this integration was to enrich our dataset, bringing in a greater diversity and quantity of examples.

- **Roboflow, platform for process computer Vision tasks**

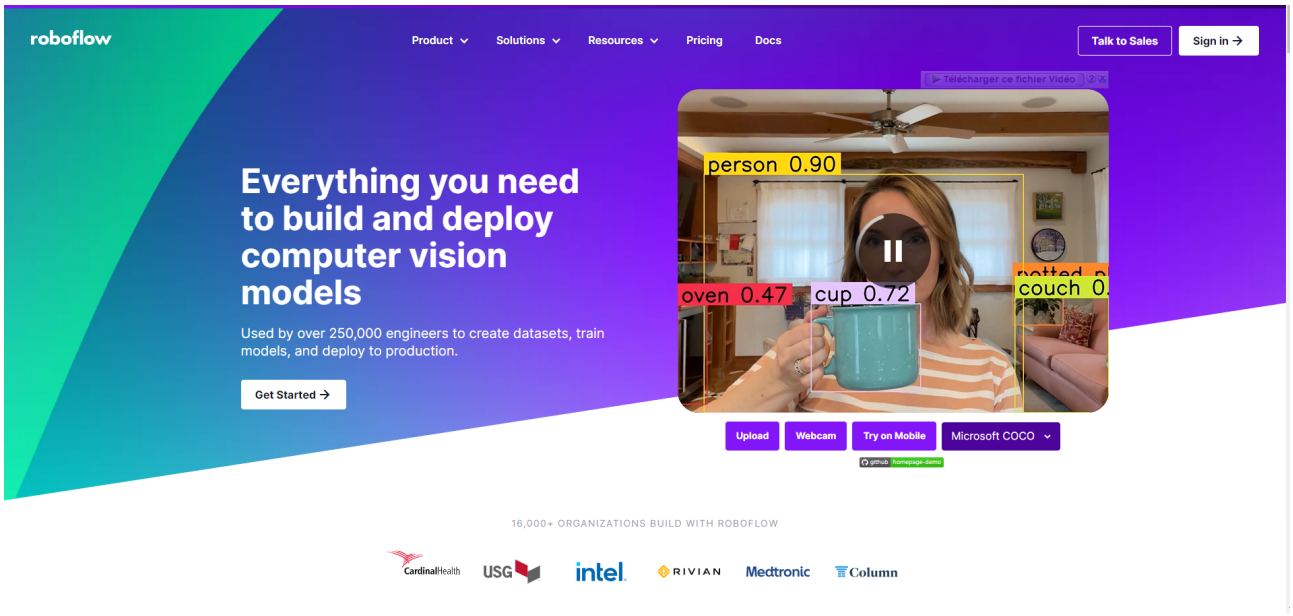

Figure 2.3: Roboflow, a tool that eases the computer vision task[11]

- Designing the Dataset Customizer

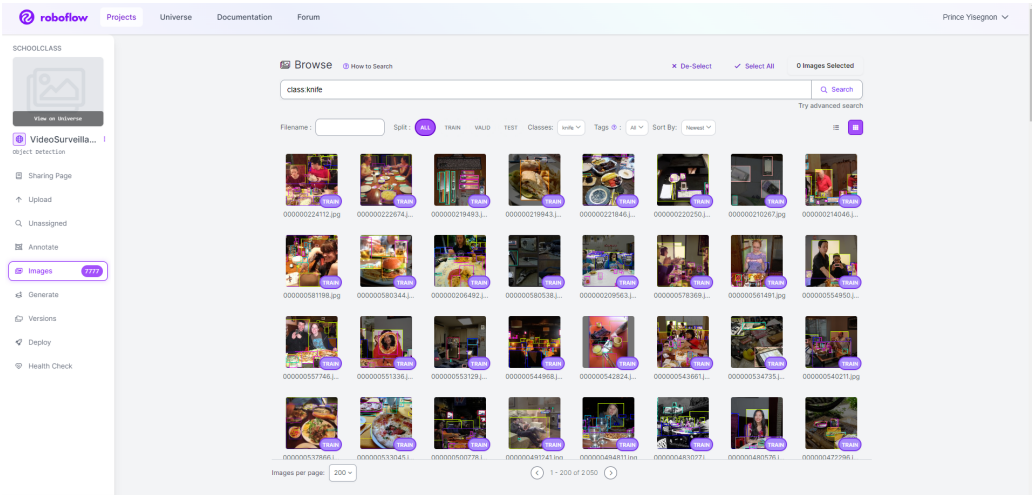

Figure 2.4: Roboflow, overview images annotated

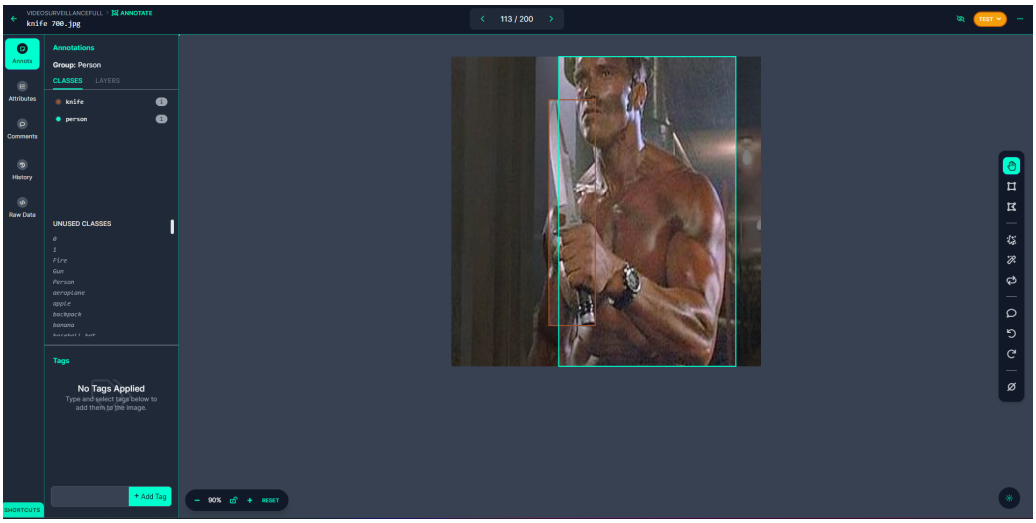

Figure 2.5: Annotation Objects

For image processing in the creation of our specific dataset for YOLO model training, we used Roboflow. This platform offers the ability to upload custom datasets, make annotations, change image orientation, resize images, adjust contrast and perform data augmentation. Roboflow can also be used for model training. As I mentioned, Roboflow also features a universal annotation conversion tool. This tool allows users to download and convert annotations from one format to another, eliminating the need to write conversion scripts for custom object detection datasets.

2.2.3 Natural Language Understanding

To facilitate interaction with our system, we have used Natural Language Understanding (NLU). This was achieved through the integration of ChatGPT API coupled with Langchain, a language processing framework. We designed a robust NLU system that enables users to input text queries, which are subsequently processed and transformed into actionable commands. This functionality greatly enhances the user experience by simplifying interaction with the system.

In this subsection, we will delve into the technical details of our NLU implementation, including the integration process, the role of ChatGPT, and how we established a seamless interaction between text queries and system commands.

```
template =  """Liste type_action : [{list_actions}]
    Liste type_objet : [{list_objets}]
    Corresponds ma question à la commande appropriée en combinant un élément de type_action et un ou plusieurs éléments de type_objet provenant des listes fournies.
    Le résultat doit être un string. Si aucune correspondance n'est trouvée, le résultat doit être 'Aucune correspondance'.
    La structure de la commande est 'objet_type_action' +' '+'objet_type_objet'."""

system_message_prompt = SystemMessagePromptTemplate.from_template(template)
human_template = "Question : {request}"
human_message_prompt = HumanMessagePromptTemplate.from_template(human_template)


1 usage  princegedeon *
def describe(request):
    chat_prompt = ChatPromptTemplate.from_messages([system_message_prompt, human_message_prompt])
    chain = LLMChain(
        llm=llm,
        prompt=chat_prompt,
        output_parser=CommaSeparatedListOutputParser()

    )
    return chain.predict(list_actions="'detecter_presence', 'compter', 'compter_par_rapport_a_une_ligne', 'programmer_evenement', 'tracker'" ,
                         list_objets="'Person', 'Cigarette', 'Fire', 'Pistol', 'Phone'", request=f"{request} ?")
```

Figure 2.6: Llms ChatGpt with LangChain

2.2.4 Integration and Real-time Enhancement

The heart of our project lies in the integration of these components and the real-time enhancement of our platform. Here, we bring together the Object Detection Model and the Natural Language Understanding system to create a unified, intelligent video surveillance solution.

We'll explore how we seamlessly integrated these modules into our system, enabling real-time processing and interaction. This section will also highlight the ways in which this integration enhances the overall capabilities of our platform, fostering intelligent, responsive video surveillance.

2.3 Modeling

2.3.1 UML Modeling

We presented three types of UML diagrams describing our system. The first put a zoom on the functionalities of our platform, it is the use case diagram. The second highlights the components in our system: it is the class diagram. The last one is a sequence diagram.

2.3.1.1 Use case diagram

As announced, this type of diagram is used to give an overall view of the functional behavior of a software system. A use case represents a discrete unit of interaction between a user and a system. It is a significant unit of work.

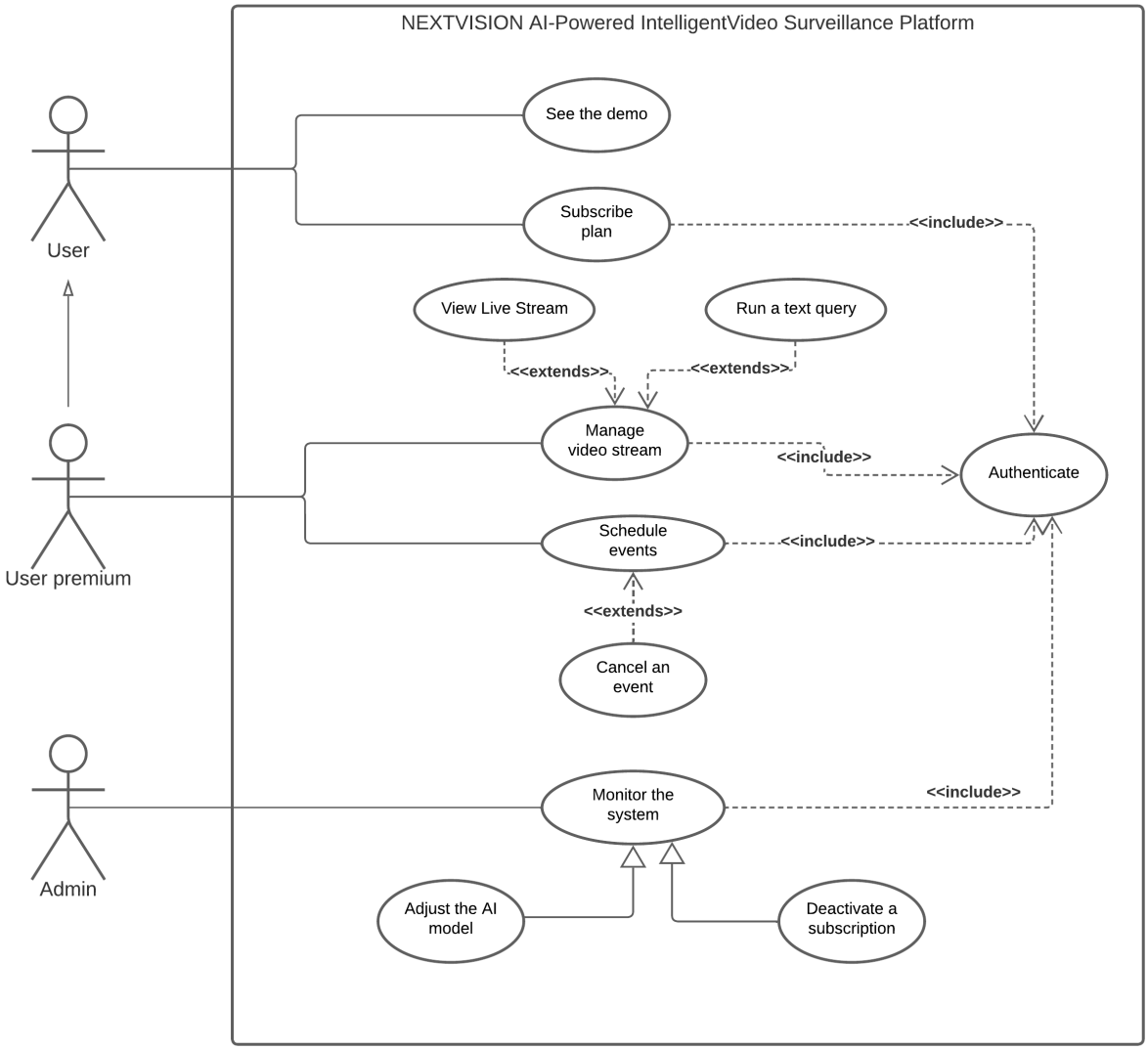

Figure 2.7: NEXTVISION use case diagram

2.3.1.2 Class diagram

In order to specify the classes involved in the system and what links they maintain between them, we use a class diagram. For development, this type of diagram is the most appropriate and gives a panoramic view of the system. *Since our platform generates class diagrams, understand that this class diagram illustrating our platform is generated by our own algorithm.* Figure 2.8 presents the first official diagram of the NEXTVISION solution.

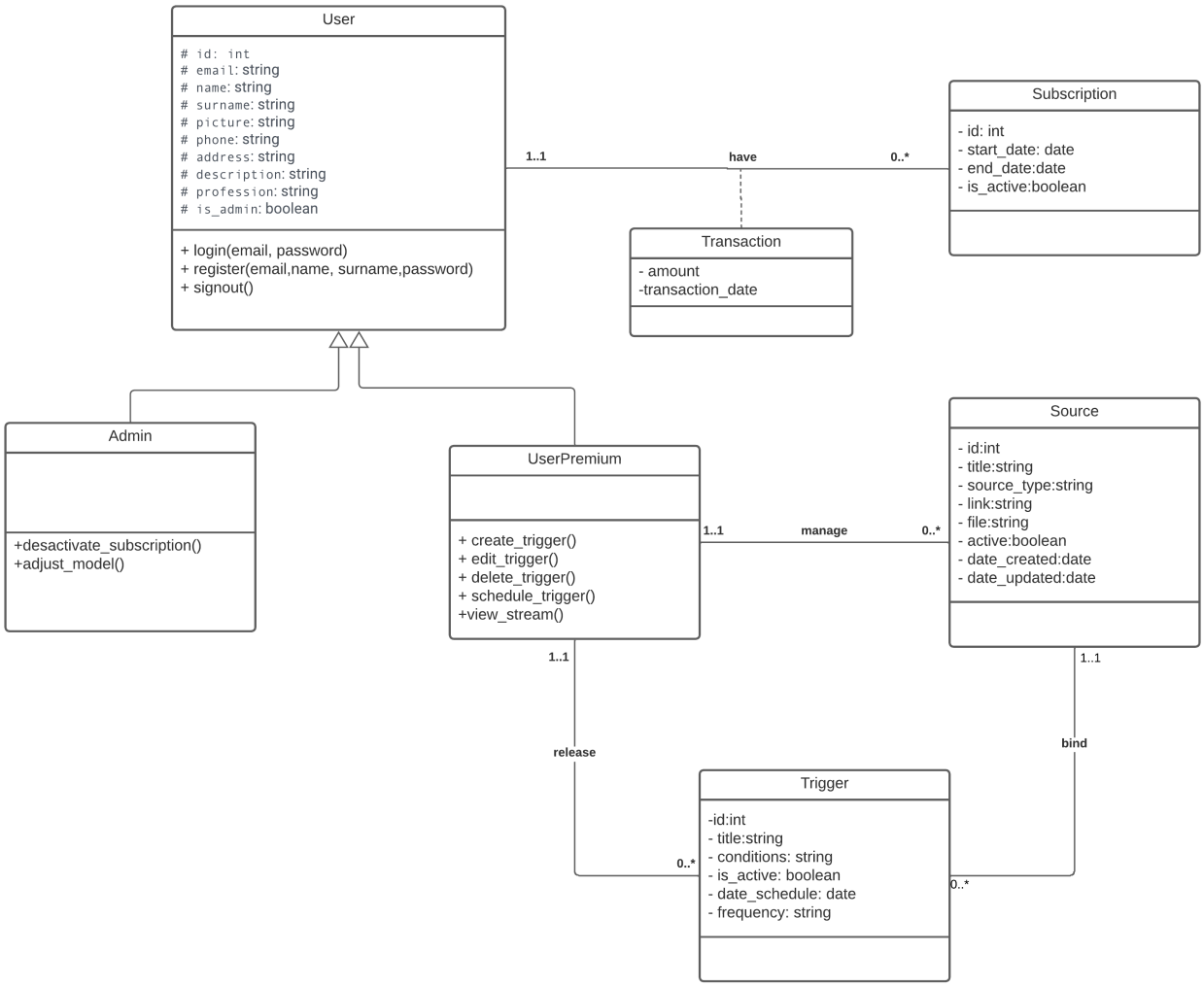

Figure 2.8: NEXTVISION class diagram

We can observe that we have seven (7) classes: User, Admin, UserPremium, Subscription, Trigger, Source, and Transaction. A user is characterized by an identifier, a name, a surname, an email, a picture, a phone, an address, a description, a profession a password, and an admin status. We have two types of users. The first one is an Admin and the second is a UserPremium. When a user joins a subscription he becomes premium. A Source is described by an identifier, a title, source_type, link or file, date of creation, and an active status. A Subscription can contain start date and end date of the subscription.

2.3.1.3 Sequence diagram

- **Modeling sequence diagram**:

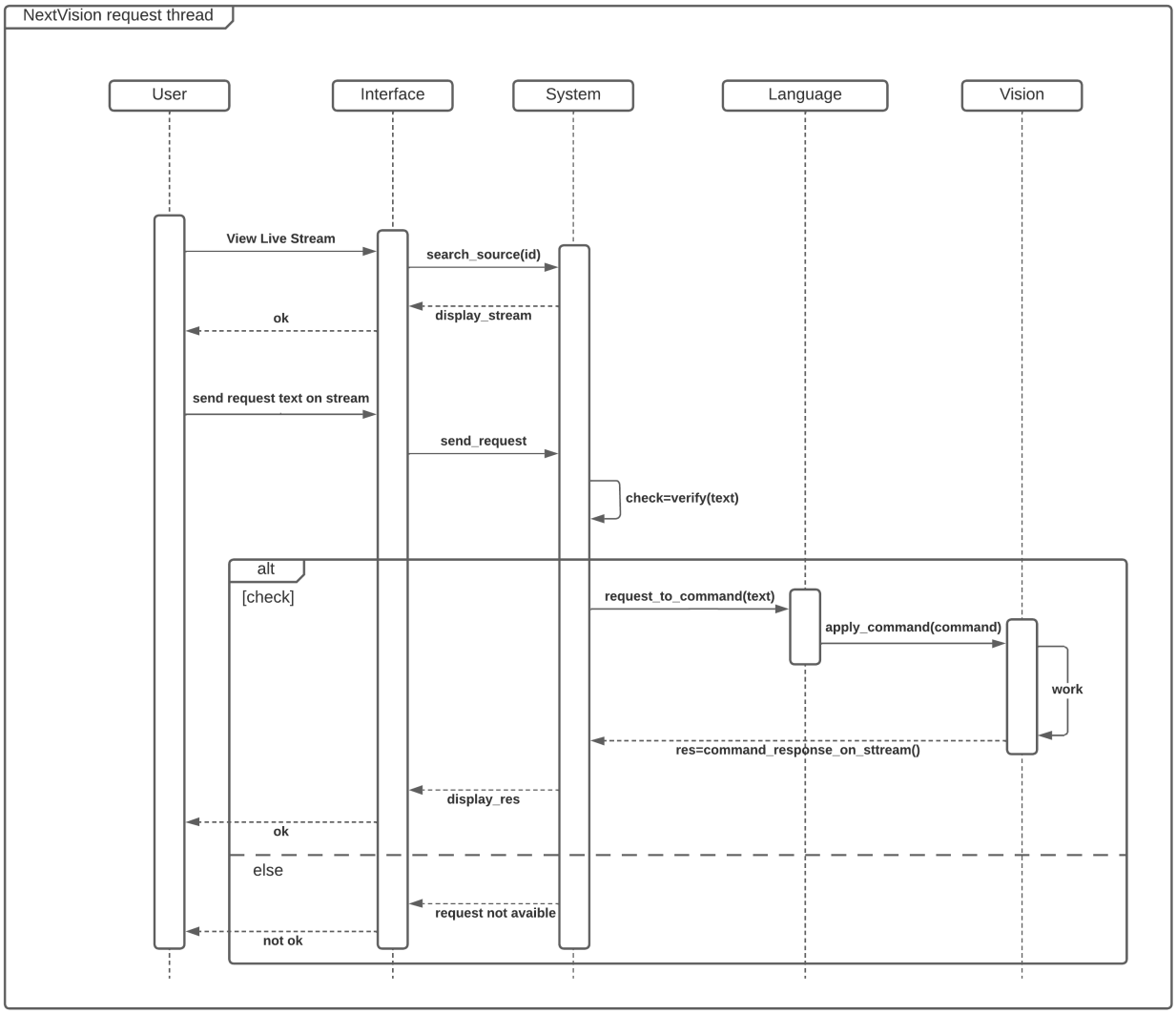

Figure 2.9: NEXTVISION modeling sequence diagram

According to 2.9, the sequence diagram involves five main modules: User, Interface (the platform), the System itself, a Language module responsible for understanding queries, and a Vision module containing all computer vision functions.

In terms of the task, when a User initiates a textual query through the Interface, requesting to view a video stream, the System searches for it and displays it. At this point, the User can write a textual query defining the task they want to execute on the Interface. This query is sent back to the System, which checks if it's valid. When the query meets all the requirements, it's forwarded to the Language module, which translates it into a command understandable by the Vision module. The Vision module executes the corresponding task, and the real-time response is sent back to the Interface. If the query is not valid, the User receives a notification on the Interface that the query is invalid and can try another one

2.3.1.4 Transition state diagram

- **Modeling transtion state diagram**:

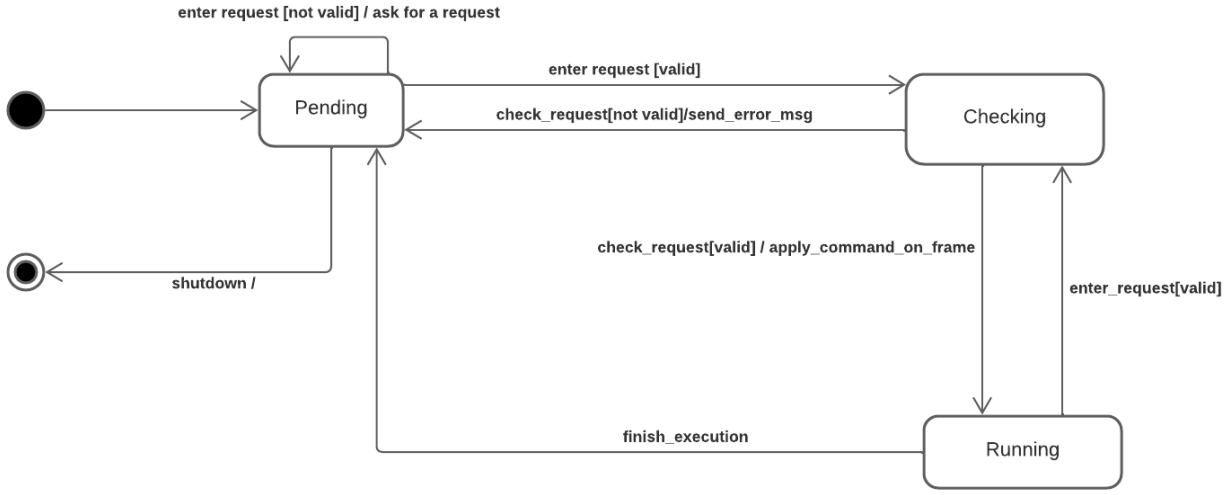

Figure 2.10: NEXTVISION Transition state diagram

The state-transition diagram2.10 illustrates the life cycle of a request within the system. Initially, the system is in the "Pending" state, waiting for a request. When a query is submitted, if it is invalid, the system remains in the "Pending" state, requesting a valid query. For a valid request, the system switches to the "Checking" state, where the request is analyzed. If the analysis reveals that the request is invalid, an error message is sent and the system returns to the "Pending" state. If the request is validated, the system applies the command and switches to the "Running" state, where the action is executed. Once execution is complete, the system returns to the "Pending" state, ready for a new request. In addition, the system can be stopped at any time, symbolizing a "shutdown" state. This cycle ensures a continuous flow of query processing, with checkpoints for validation and error handling.

Conclusion

NextVision brings added value by simplifying complex video surveillance tasks through advanced natural language processing. It seamlessly incorporates video loading and uploading from various sources, such as existing libraries, IP cameras, and camera systems. Users can issue real-time commands and receive instant responses, elevating the project's capabilities significantly. Our architectural framework, built upon Django, Next.js, and WebSocket technologies, provides a solid and adaptable foundation.The following chapter is devoted to the presentation of the results of the implementation of our platform .

Chapter 3

Results and discussion

Introduction

The previous chapters allowed us to make a state-of-the-art, to present the tools and methods we used to obtain our model. In this chapter, we present the obtained results.

3.1 Overview of the solution

Our solution revolves around three key components. Firstly, we developed our vision model with object detection and tracking tasks, which constitute the core functionalities of our application. Next, we containerized our model using Docker. Subsequently, we expose our model processing pipeline through an API, which encompasses all the logic of our system, including authentication and real-time video stream management with our Vision and Language modules. To harness the potential of our API, we have created an interactive web platform that enables users to query the API, thus facilitating the consumption of the application.

3.2 Result

3.2.1 Vision module development

The development of our vision module involves creating a system capable of processing images (frames) and detecting objects within them. To do this, we are using YOLO (You Only Look Once) technology, initially trained to recognize almost 1000 different classes. We have adapted and refined this model to focus specifically on our five target classes. The use of Roboflow was crucial during the data collection and annotation phases. This platform enabled us to manage this data efficiently, simplifying the development process of our module. The following steps cover the entire process, from initial data collection to the final development of the vision module.

Preprocessing
Before starting the training, we must do the preprocessing steps such as putting all the images at the

same size, normalizing the pixels, splitting data into test, train, and validation categories, and finally exporting to the appropriate format to start the training. with YOLOv8.

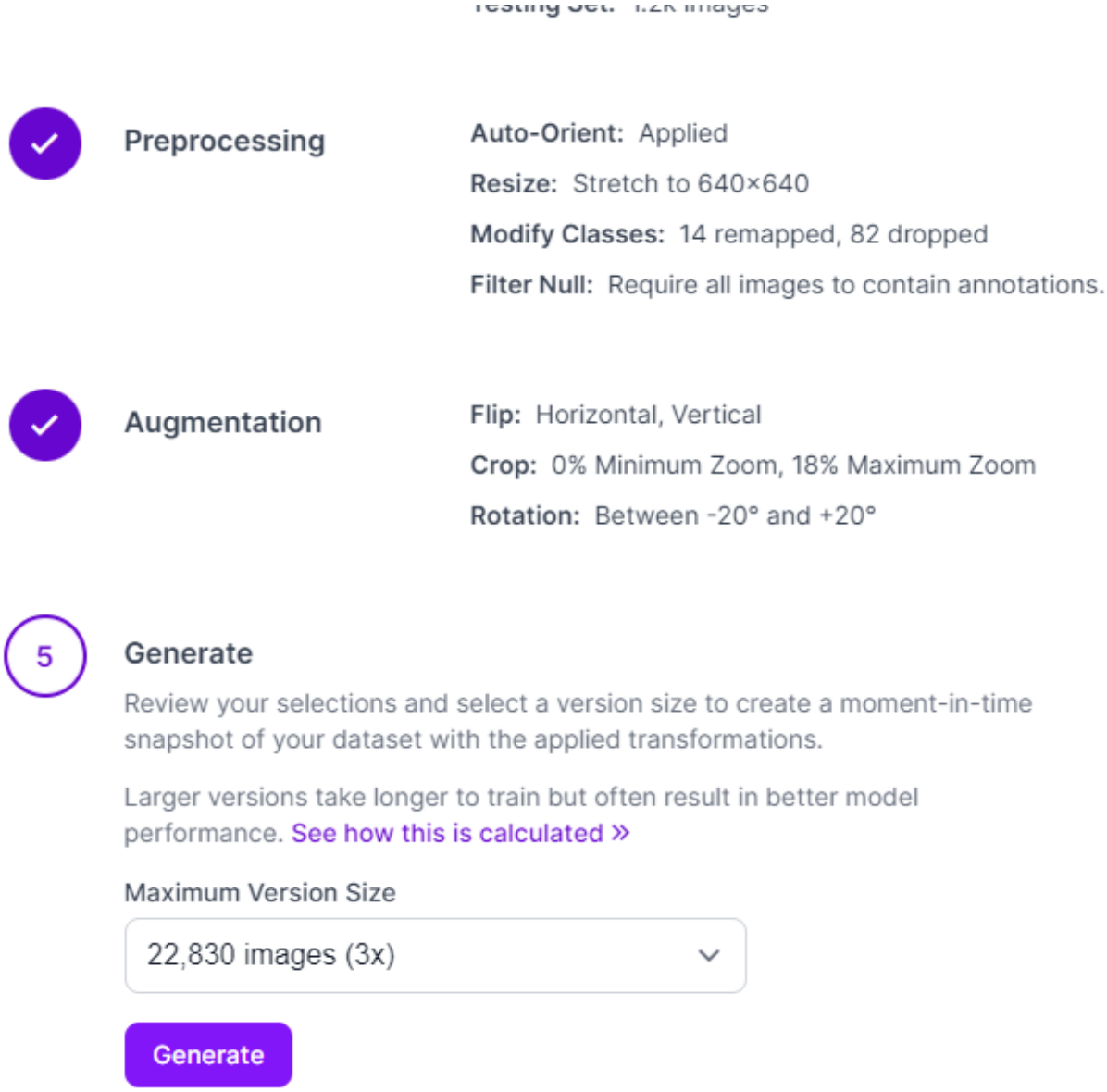

Figure 3.1: Step preprocessing data

Basically, we had 9902 images. To predict 5 classes it was enough. We did what we call **Data Augmentation** by creating variants of the same images by applying rotations and flips. And in the end we have 17,020 images.

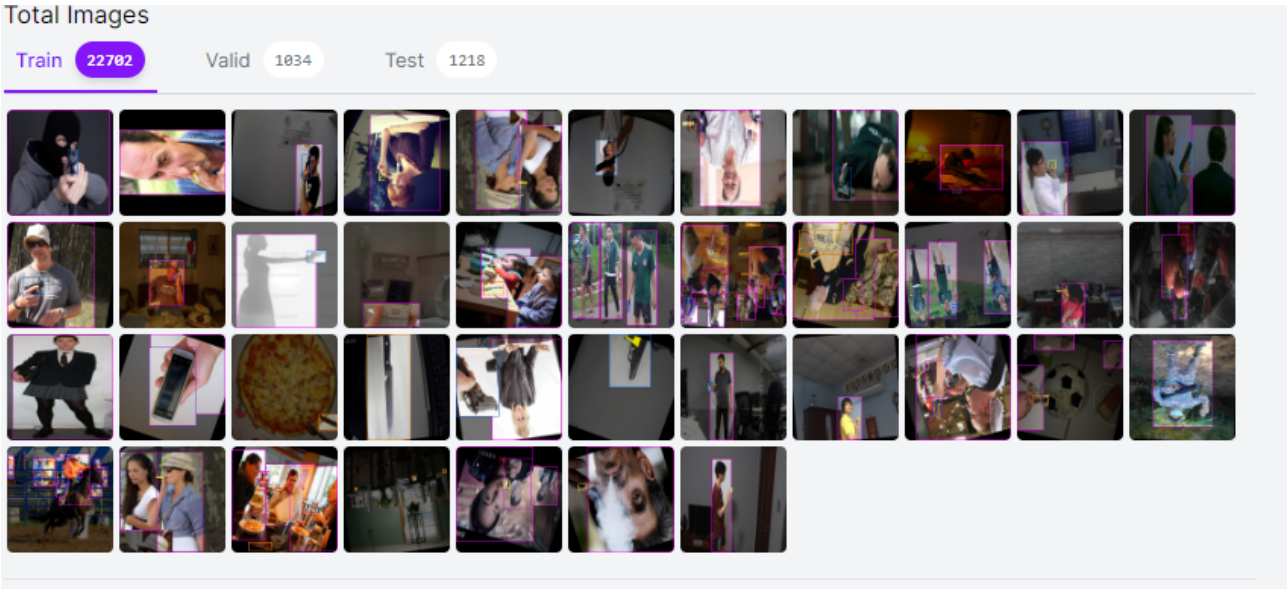

Figure 3.2: Dataset custom obtained after transformation

Training objects detections models

In this critical phase, we engaged in the training of our object detection models using YOLOv8. This step represents the heart of our efforts to build a powerful and accurate system for intelligent video surveillance.

data.yaml × labels.jpg

```
names:
- cigarette
- gun
- knife
- person
- phone
nc: 5
roboflow:
  license: Public Domain
  project: videosurveillanceall
  url: https://universe.roboflow.com/schoolifri/videosurveillanceall/dat
  version: 1
  workspace: schoolifri
test: /content/datasets/VideoSurveillanceAll-1/test/images
train: /content/datasets/VideoSurveillanceAll-1/train/images
val: /content/datasets/VideoSurveillanceAll-1/valid/images

```

Figure 3.3: Data format before training

After exporting our data see 3.3 , we launched the training notebook

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8m.pt data=/content/datasets/VideoSurveillanceAll-1/data.yaml epochs=175 imgs:
```

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
144/175	6.9G	0.709	0.5319	1.069	67	640: 100% 1419/1419 [02:54<00:00, 8.14it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.818	0.743	0.778	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
145/175	6.9G	0.7061	0.5278	1.067	72	640: 100% 1419/1419 [02:53<00:00, 8.17it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.812	0.746	0.778	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
146/175	6.88G	0.7015	0.5238	1.062	66	640: 100% 1419/1419 [02:54<00:00, 8.15it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.813	0.745	0.778	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
147/175	6.9G	0.6981	0.5228	1.062	94	640: 100% 1419/1419 [02:53<00:00, 8.16it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.817	0.747	0.778	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
148/175	6.9G	0.6949	0.5167	1.06	61	640: 100% 1419/1419 [02:53<00:00, 8.18it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.827	0.745	0.778	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
149/175	6.9G	0.6901	0.5163	1.059	46	640: 100% 1419/1419 [02:54<00:00, 8.15it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.823	0.744	0.777	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
150/175	6.89G	0.6881	0.5099	1.056	69	640: 100% 1419/1419 [02:54<00:00, 8.15it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.821	0.747	0.777	0.539				
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
151/175	6.9G	0.6872	0.5104	1.054	77	640: 100% 1419/1419 [02:53<00:00, 8.17it/				
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 33/33 [00:05<00:00,				
all	1034	2052	0.82	0.747	0.777	0.538				

Figure 3.4: a notebook training YOLOv8

We started the training with Google Colab Pro[9], on a machine supporting 40 GB GPU A100 and 80 GB RAM.

According to 3.4, over time, our model learns to identify our 5 objects on images by acquiring experience. The training lasted 8 hours and over 175 iterations.

Evaluate the performance of the model

Once the training was completed, we rigorously evaluated the model's performance. Real-world testing was conducted to ensure that the model effectively identified security-related items under different conditions and scenarios.

Performance Improvement with Additional Data and Epoch Adjustment

After the initial model training and evaluation, we recognized the opportunity to enhance performance further. We conducted two additional experiments, modifying key parameters and dataset distribution to observe their impact .

1. **Experiment 1: First Train**

   In the first experiment, we fixed the number of training epochs to 200 on 15.000 images dataset

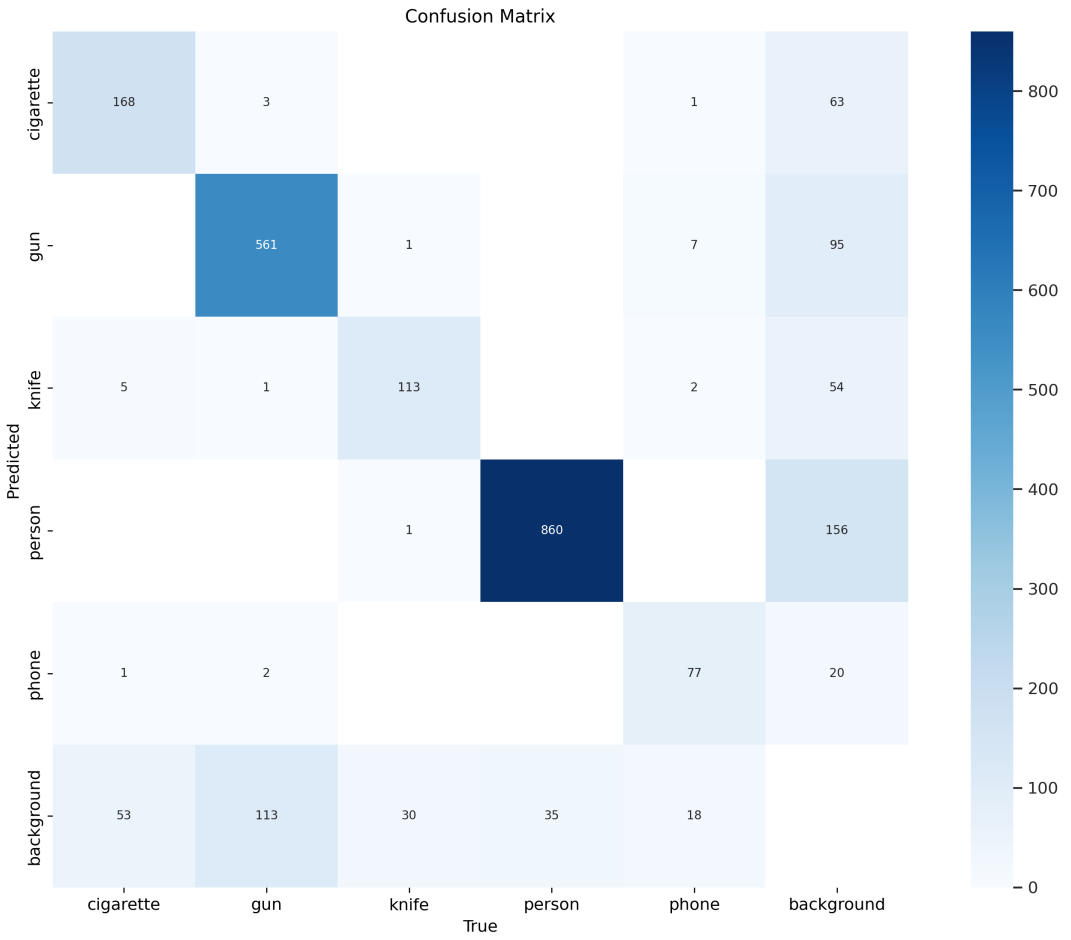

Figure 3.5: Confusion matrix after first train

2. **Experiment 2: Increasing Data Volume and dataset distribution**

   Recognizing the impact of data on model performance, we introduced additional labeled images, with a particular focus on the "knife" class. This deliberate inclusion of more knife-related data aimed to improve the model's ability to accurately detect and classify knives.

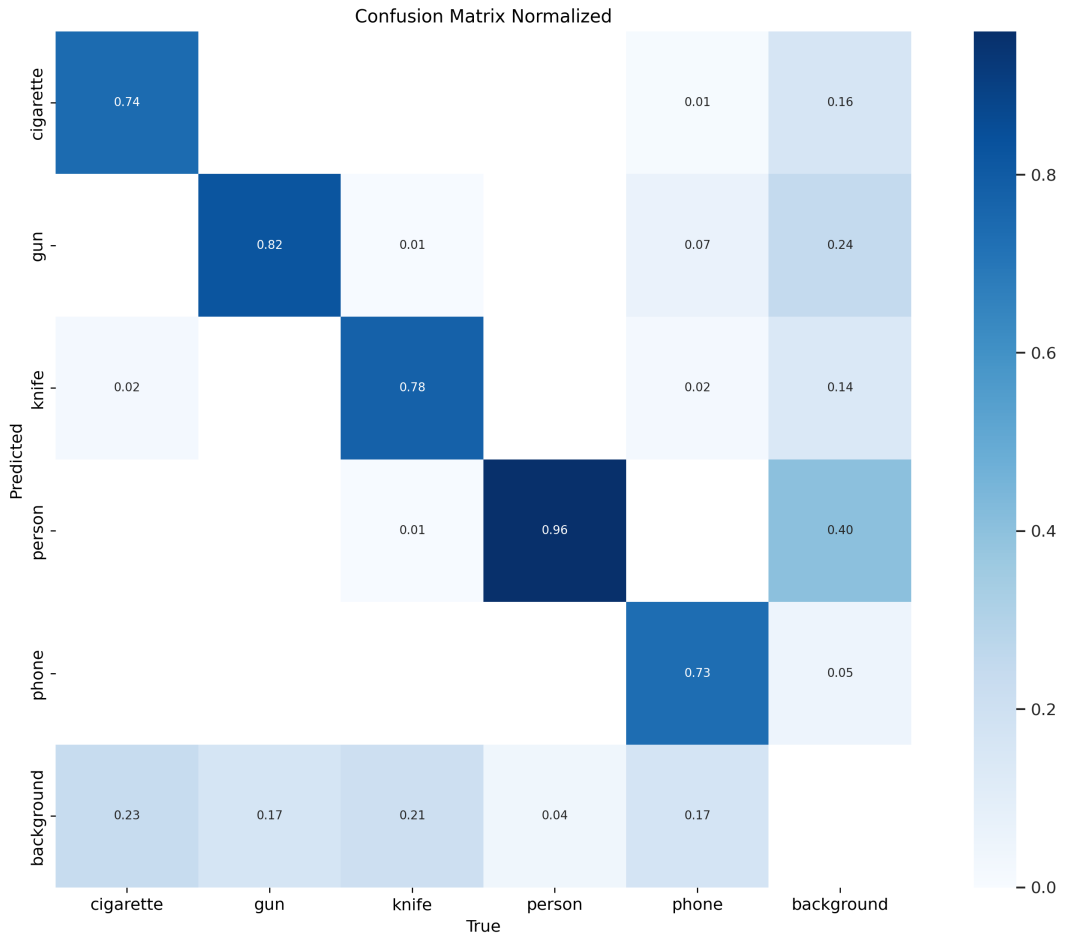

Figure 3.6: Confusion matrix after augmenting data, with a focus on the "knife" class

Observations and Conclusion

The comparison of the two confusion matrices highlights the positive impact of these adjustments. The second experiment, which involved augmenting the dataset with more knife-related images, resulted in a notable enhancement in the model's performance. The increased data volume contributed to a more robust and accurate detection of the "knife" class, ultimately improving the overall effectiveness of the YOLOv8 model.

The confusion matrix you are referring to graphically represents the performance of a classification model, specifically YOLO, an object detection algorithm, across five distinct classes: cigarette, gun, knife, person, and phone. The rows labeled "Predicted" indicate the model's predictions, while the columns labeled "True" reflect the actual object classes in the test images. Each matrix cell displays the normalized rate of predictions for a predicted-true class pair. For instance, the model accurately predicted "gun" with a rate of 0.82, meaning that in 0.82 of the instances where the model predicted "gun," it was correct.

The **background**[59] class is likely included to account for scenarios where none of the specific objects of interest are present in the image, or to identify image regions that don't contain any significant objects. It's a common aspect in object detection models to differentiate between areas of interest and non-significant regions. The matrix values for the background prediction suggest that the model occasionally has to determine whether an image contains one of the objects of interest or if it's merely part of the background.

Result graphs

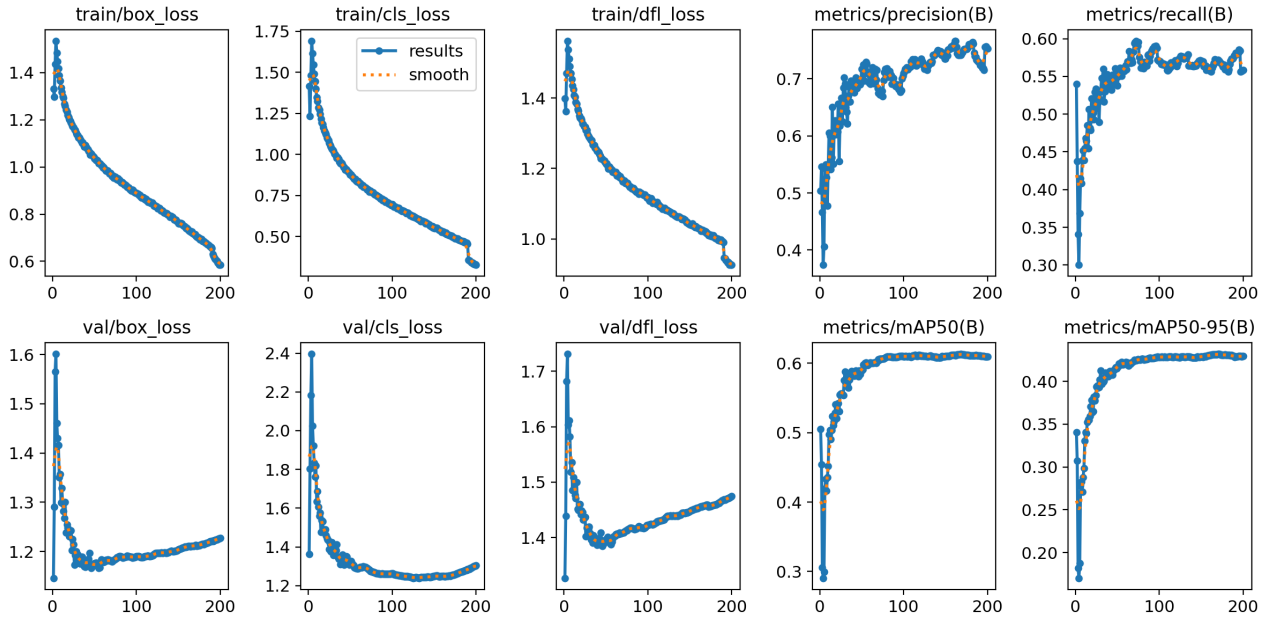

Figure 3.7: confuxion matrix after training YOLOv8

According to 3.7, our model improves as training progresses by analyzing the graphs of precision, recall, and accuracy metrics as a function of epochs. Similarly, we have the loss function that decreases as iterations progress. Our model is training effectively.

Curve graphs

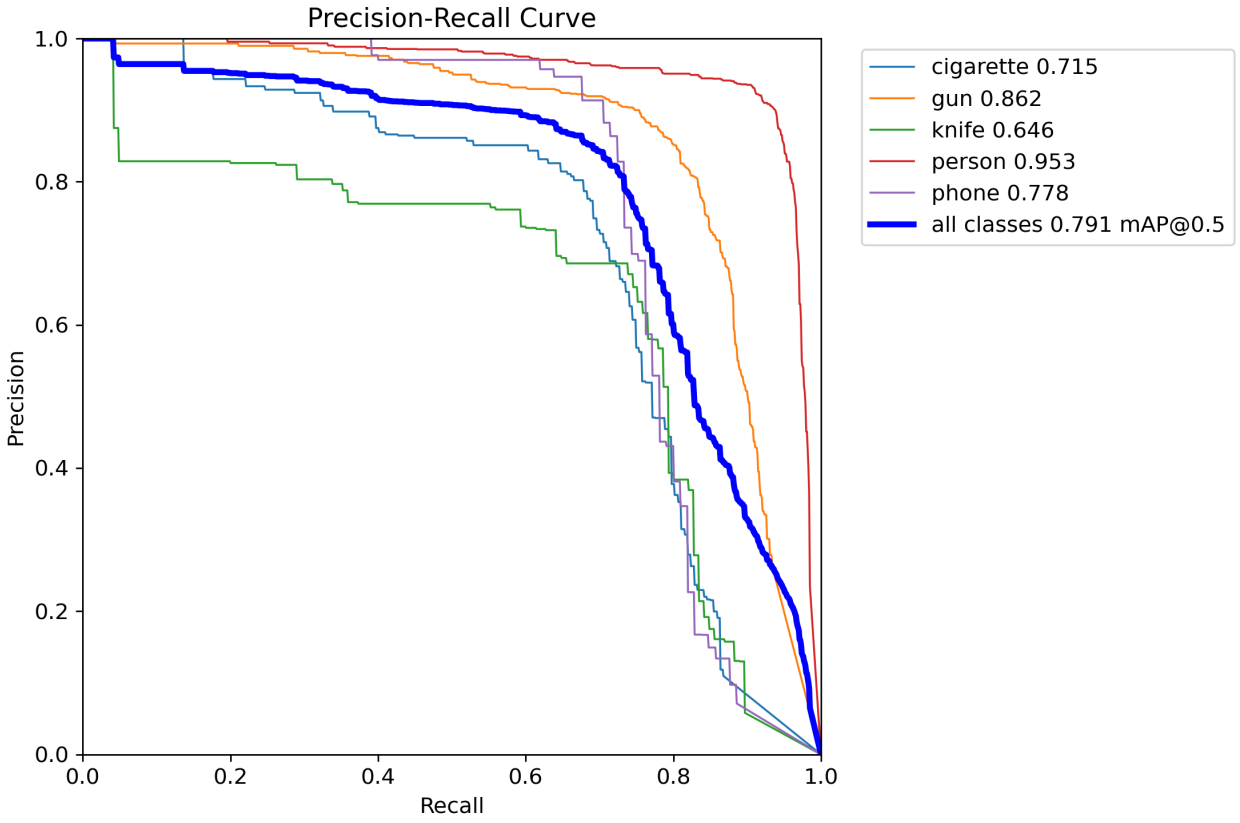

Figure 3.8: Precision-Recall Curve

According to 3.8, The graph is a Precision-Recall curve for a model detecting five classes: cigarette,

gun, knife, person, and phone. Each line shows the trade-off between precision (correct positive predictions) and recall (finding all positives) for a class. The Average Precision (AP) for each class and the mean AP (mAP@0.5) for all classes at an Intersection over Union (IoU) threshold of 0.5 are given, indicating overall model accuracy. 'Person' has the highest AP, while 'knife' has the lowest, with an overall good model performance shown by the mAP of 0.791.

Overview of model predictions on a batch of images

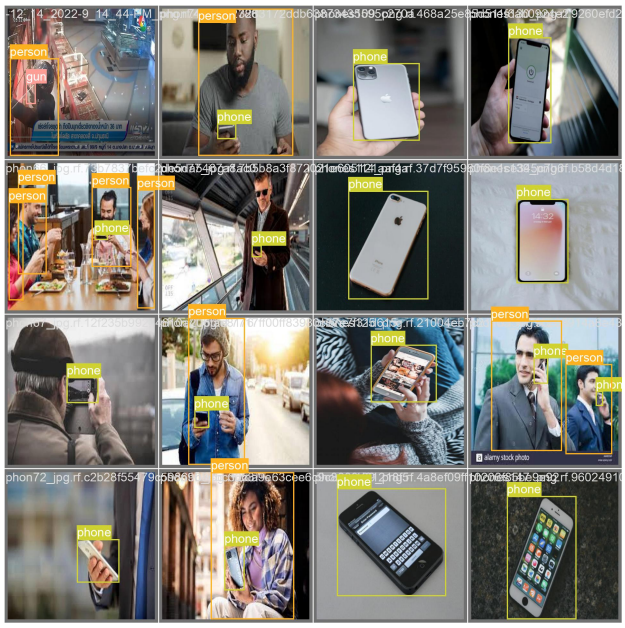

Figure 3.9: Model prediction on batch test 1

Figure 3.10: Model prediction on batch test 2

Figure 3.11: Model prediction on batch test 3

In conclusion, our model demonstrates reasonable performance, successfully identifying individuals, phones, and cigarettes within images. While it faces challenges in accurately detecting knives, especially from certain angles, we are overall satisfied with its performance, particularly given that this is our initial iteration and considering the limitations of our dataset.

3.2.2 API

- **API /nextvision endpoint**
  Figure 3.12 illustrates the API endpoints of NextVision in the Swagger documentation. Within this context, we have an endpoint dedicated to managing video sources and all the authentication routes that utilize JWT tokens

Figure 3.12: NEXTVISION API source endpoint

- **API /uml/class-diagram-obj result**
  Figure 3.13 shows an API call of our model.

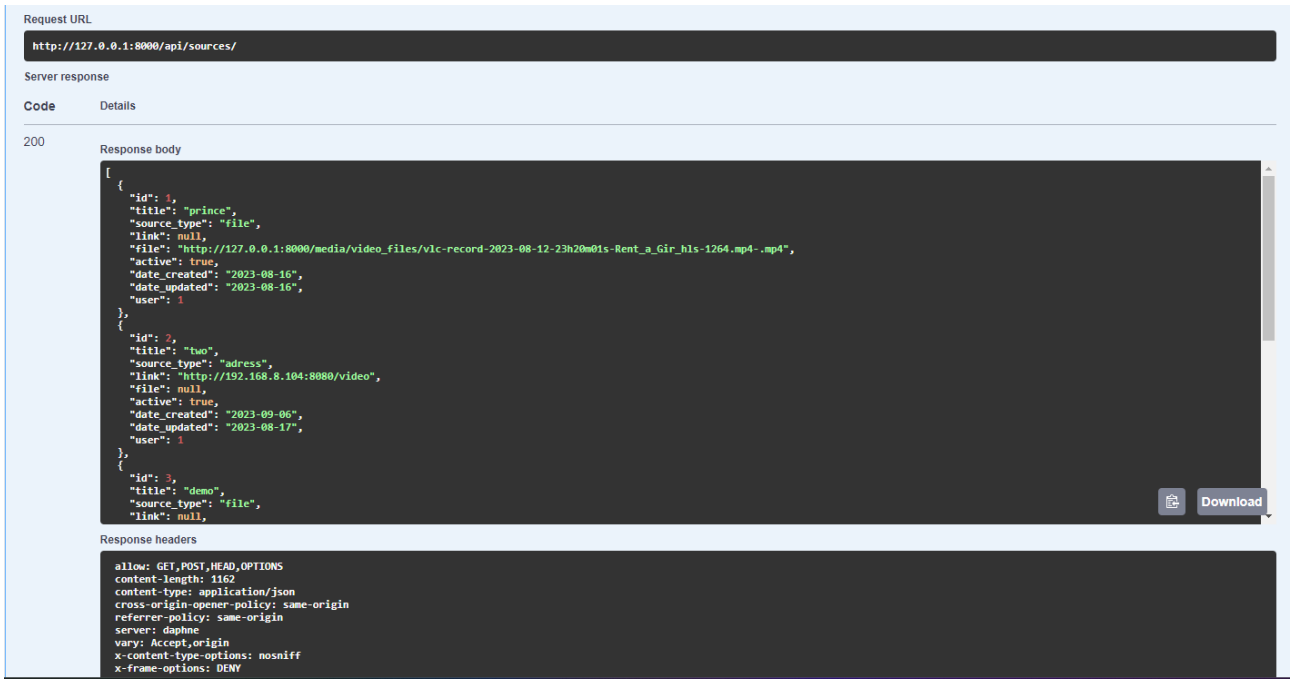

Figure 3.13: NEXTVISION API source result

- **API websocket test**

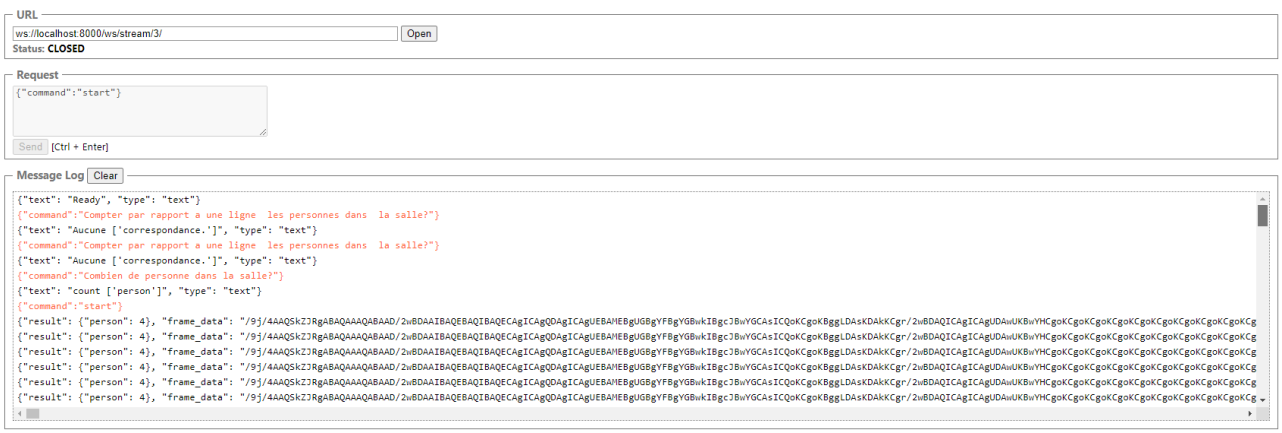

Figure 3.14: NEXTVISION websocket test

Figure 3.14 shows a test of the websocket. At the beginning, the Ready message is sent by the system to show that everything is loaded. Then the user types this request which is converted into a command by the system and as soon as he puts start the processing begins

3.2.3 Interfaces

- **Landing page**
  The Figure 3.15 shows the NEXTVISION landing page screenshot.

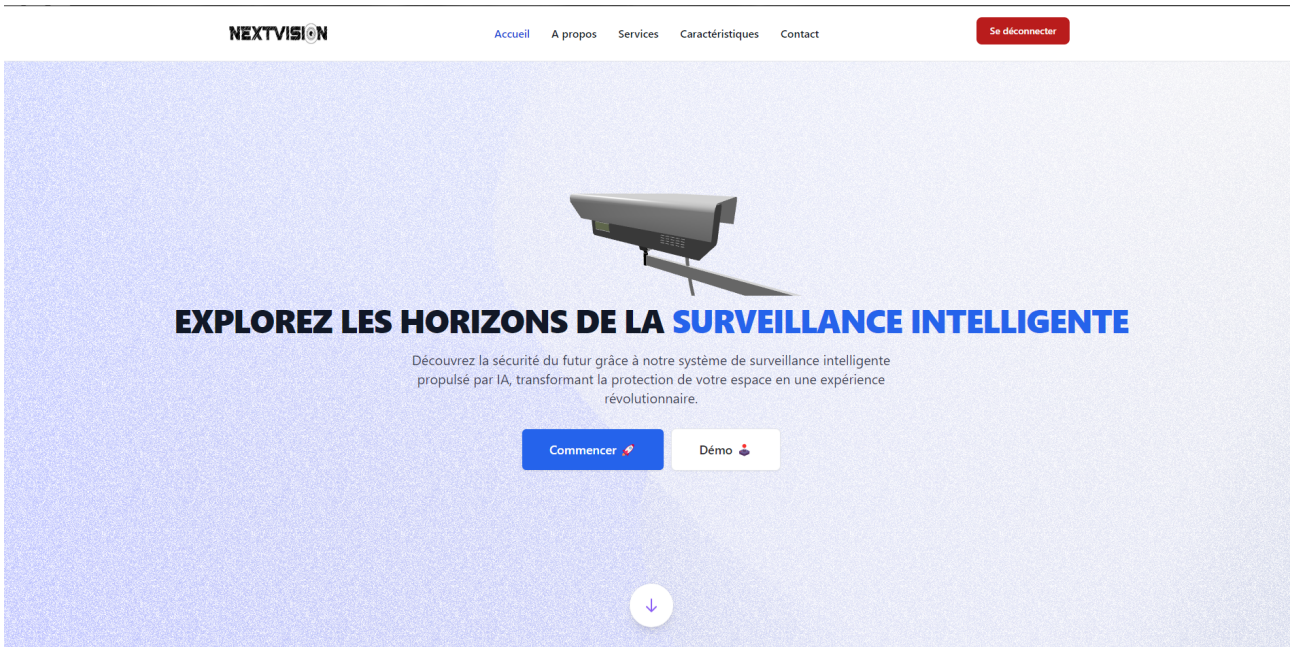

Figure 3.15: NEXTVISION web application landing page

- **Registration**
  The interface presented in figure 3.16 allows people to create an account. To do this, they must specify their email address, password, first name, and last name. In order to ensure the security of user accounts, passwords are encrypted before being saved in the database.

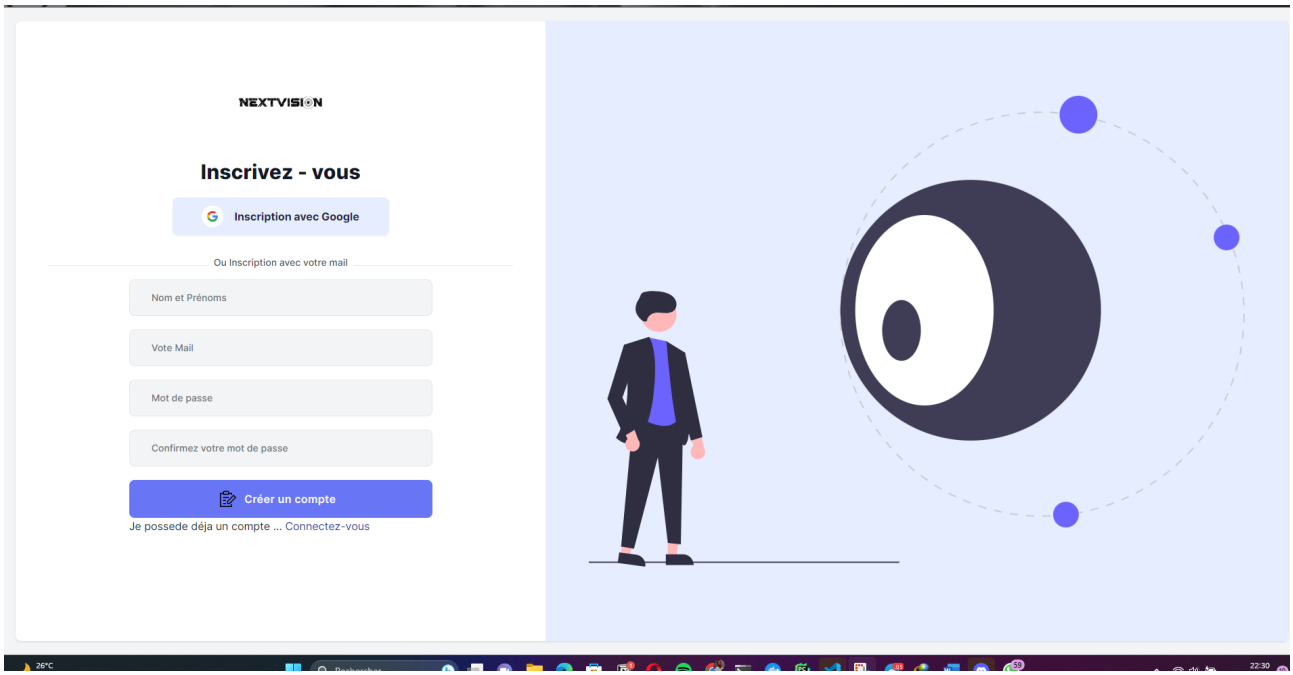

Figure 3.16: NEXTVISION Register page

- **Login**
  The interface presented in figure 3.17 allows the user to log in once the email and password tuples are verified. Our API, therefore, returns an access token that allows performing actions that require authentication in the web application.

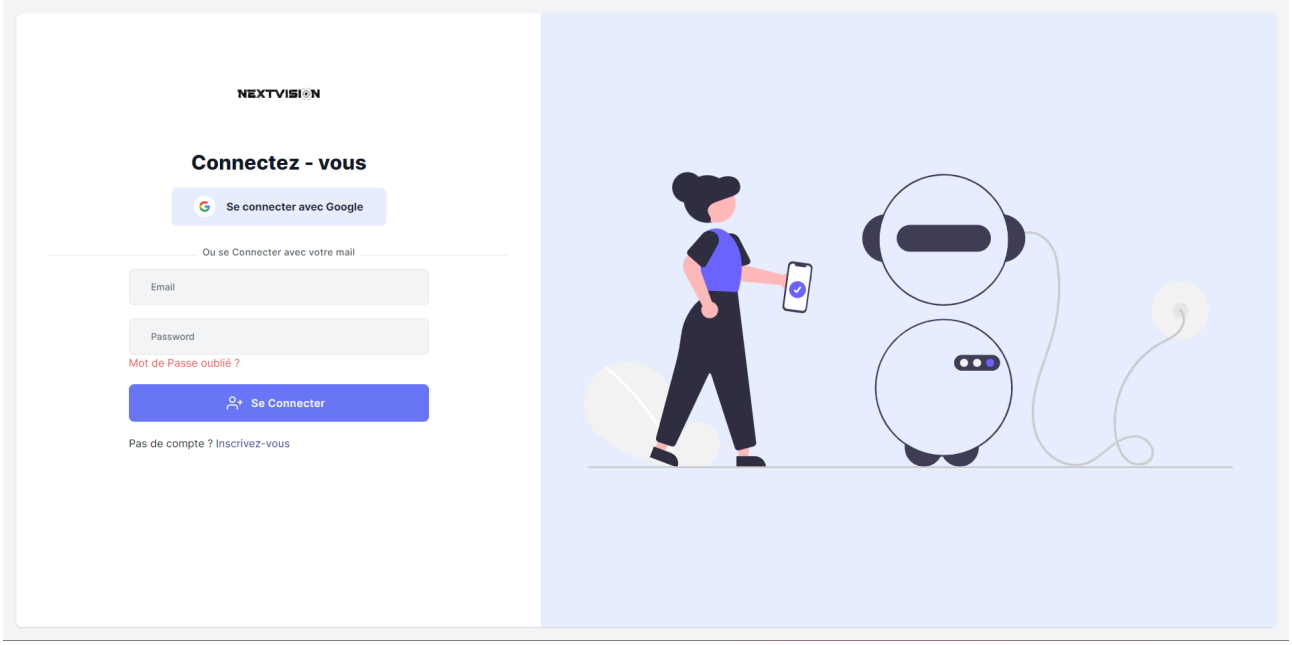

Figure 3.17: NEXTVISION Login page

- **Dashboard**
  The interface described in Figure 3.18 represents our dashboard, the main page where users can view their statistics, create video sources and streams, manage them, and even delete them. It also allows for advanced filtering options.

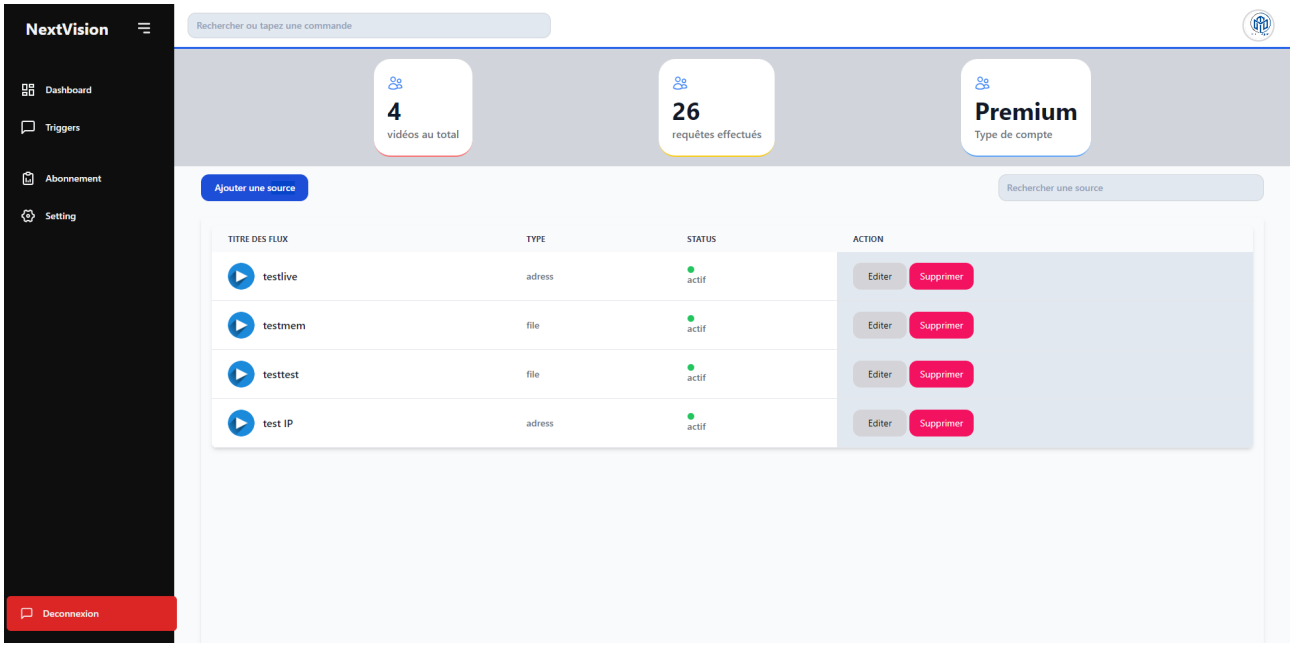

Figure 3.18: NEXTVISION Dashboard page

- **Playground**
  The interface depicted in Figure 3.19 represents the playground page for each stream. It consists of a live video stream and a chat interface. The chat is integrated with WebSocket, combining both NLP functionality and vision processing. When the user submits a text query, it is converted into a command, interpreted in the backend by the vision module, and then sent back to

the interface via WebSocket.

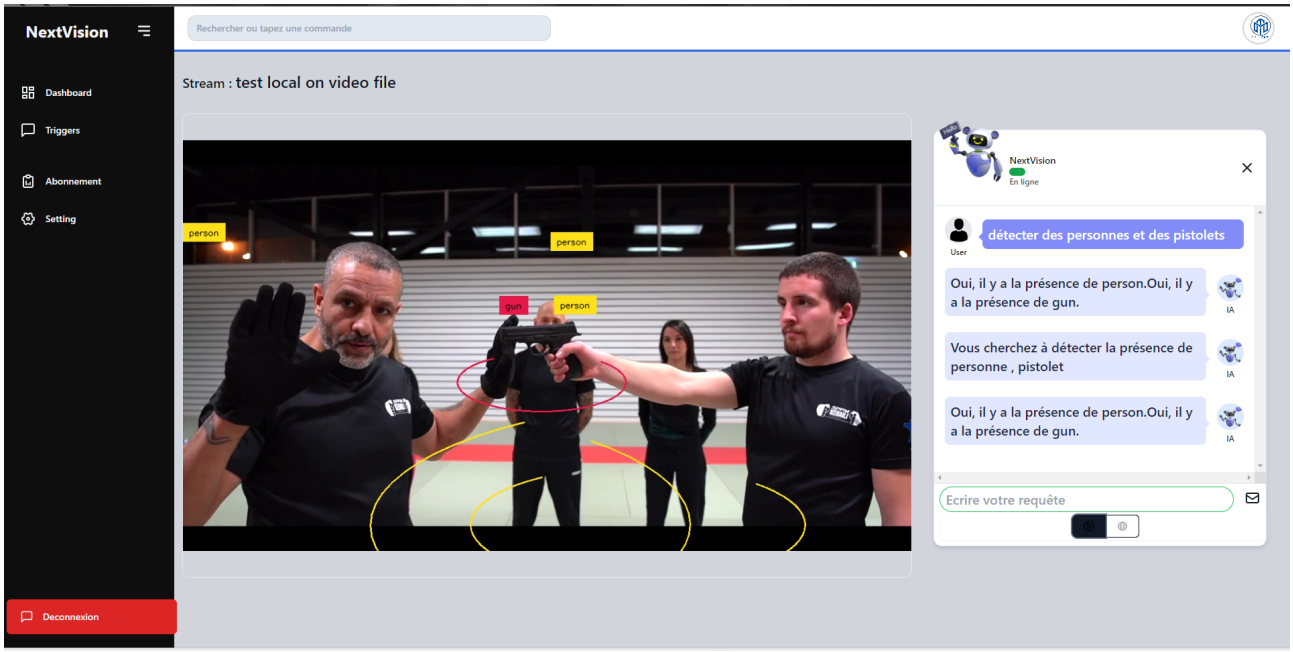

Figure 3.19: NEXTVISION Playground page

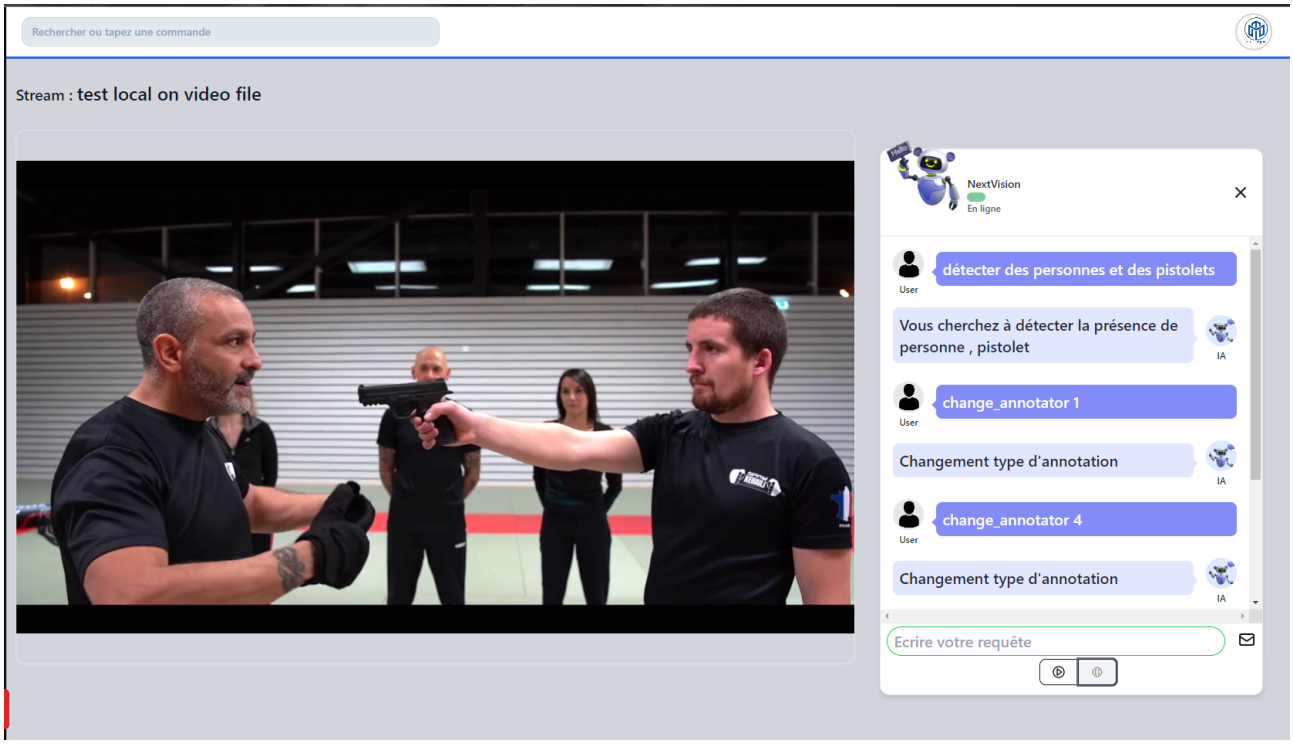

Figure 3.20: NEXTVISION Playground 2 page

3.3 Discussion

3.3.1 Technical specifications

NetVision FPS performance for real-time video processing

The FPS (frames per second) performance of our NetVision system for real-time processing of video streams depends on several key factors. These include hardware configuration, such as GPU and CPU power, and the resolution of the images processed. With current hardware configurations, we observe an average performance of around 30 FPS.

It's important to note that this average may vary depending on the specific characteristics of each configuration and the complexity of the scenes processed.

To achieve this performance, we recommend the following minimum specifications:

- GPU with a minimum RAM of 4 GB.
- Total system RAM of at least 8 GB.

3.3.2 Overall difficulty

It's important to note that the pre-trained model we used wasn't performing optimally, primarily due to limited data samples. However, our model's performance was already quite respectable. Integrating the vision and language modules within the WebSocket presented some challenges that led us to conduct further research in this domain. Simultaneously streaming frames while processing user queries restricted the flexibility we aimed for. Moreover, we didn't find a readily available dataset, so we embarked on annotating nearly 10,000 images, which was a time and resource-intensive process. These were the primary challenges we encountered.

3.3.3 Evaluation method

We connected various video sources, including both MP4 files and IP camera addresses, and initiated the streaming process. During the continuous WebSocket streaming, we tested several queries through the chat interface. We were able to launch multiple queries simultaneously for different sources, enabling multi-camera surveillance. The performance was quite satisfactory, and we were genuinely impressed with the results.

3.3.4 Limits

The video surveillance intelligence application we have developed and presented demonstrates strong performance, but it's not without limitations. One of these limitations pertains to the system's knowledge base, which is currently limited to identifying just five objects, making it less versatile for broader applications. Extending this knowledge base is a necessary future step. Another limitation relates to the system's ability to process complex sentences, as it may not fully extract objects or tasks implicitly. Furthermore, while the system continuously streams video, it currently lacks the capability to rewind, fast forward, or pause the stream to pose a query and wait for a result based on

current frames. Additionally, our platform currently requires an internet connection and that cameras be within the same network for access. These limitations must be addressed to enhance the accuracy and performance of our application.

Conclusion

This chapter provides an overview of our research findings. We initially presented the results obtained from our work and then proceeded to assess both the achievements and limitations of our platform. Our API, which implements the core functionality of our model and authentication processes, is made accessible through endpoints. Additionally, we introduced the web application that we developed, which interacts seamlessly with our API to manage video sources and WebSocket communications. This application empowers users to oversee video surveillance through textual prompts. Although we have successfully met our objectives, our system evaluation unveiled imperfections and weaknesses, particularly concerning the training knowledge base. These areas require attention and improvement.

General conclusion

NextVision combines computer vision and natural language processing to deliver an exceptional user experience. The utilization of YOLOv8, a state-of-the-art object detection model, has empowered us to create a reliable detection system capable of recognizing critical objects such as firearms, fires, and more. We meticulously curated a custom dataset through Roboflow, ensuring a good performance of our model. Our natural language understanding system, built upon ChatGPT, enables users to interact with the system intuitively and naturally by submitting textual queries that are seamlessly converted into commands. NextVision represents a significant advancement in the realm of intelligent video surveillance, opening new possibilities for enhanced security and efficient monitoring.

However, there are exciting opportunities for future work. Throughout this thesis, we grappled with the global challenge of making a machine understand and interpret its surroundings. We took a rule-based approach for our object detection and natural language processing model. While this approach provided valuable insights, our model isn't performing at its best yet. To make the tool work better in the future, we have a few promising ideas. One option is to gather and label more data, which would allow our model to recognize a wider range of objects. Another approach to consider is Visual Captioning, which involves providing more detailed scene descriptions. We can also explore using advanced computer vision tools used by experts to build more powerful features. Improving our algorithm's ability to understand what's in an image is crucial. Additionally, we can use event-triggered actions based on specific conditions through queues to enhance security performance.

Bibliography

[1] Pascal Roques. "*UML 2 par la pratique*", Eyrolles, 2011.

[2] Laurence Moroney. "*AI and Machine Learning for Coders*", O'REILLY, 2020.

[3] Aurelien Géron. "*Hands-on Machine Learning With Scikit-learn, Keras, and Tensorflow* ",Third Version, O'REILLY, 2022.

[4] Rajeev RATAN. "*Modern Computer Vision*", 2022.

Webography

[5] The false and expensive miracle of video surveillance, https://www.cairn.info/revue-apres-demain-2010-4-page-28.htm, Last accessed: August 25, 2023.

[6] Artifical Intelligence, https://en.wikipedia.org/wiki/Artificial_intelligence, Last accessed: August 25, 2023.

[7] Machine Learning, https://datascientest.com/machine-learning-tout-savoir, Last accessed: August 25, 2023.

[8] Computer Vision, https://datascientest.com/computer-vision, Last accessed: August 25, 2023.

[9] Google collab, https://colab.google/, Last accessed: August 25, 2023.

[10] Deep Learning, hhttps://datascientest.com/deep-learning-definition, Last accessed: August 25, 2023.

[11] Roboflow, a tool that eases the computer vision, https://roboflow.com/, Last accessed: August 25, 2023.

[12] The world's largest collection of open source computer vision datasets and APIs, https://universe.roboflow.com/, Last accessed: August 25, 2023.

[13] Application of Artifical Intelligence Machine Learning in Real Life, https://www.researchgate.net/publication/340684782_Application_of_Introduction_Artificial_Intelligence_Machine_Learning_in_Real_Life, Last accessed: September 09, 2023.

[14] Whats is Neural Network, https://medium.com/@itsmescottb123/what-is-a-neural-network-a8cfd5b18dc0, Last accessed: September 10, 2023.

[15] Python Language, https://fr.wikipedia.org/wiki/Python_(langage), Last accessed: November 29, 2023.

[16] Understanding CNN for Image Classification, https://medium.com/@debjyoti.banerjee.201/understanding-cnn-for-image-classification-b442769878da, Last accessed: September 10, 2023.

[17] Convolutive neural network, https://en.wikipedia.org/wiki/Convolutional_neural_network, Last accessed: November 29, 2023.

[18] Simple Online And Realtime Tracking, https://arxiv.org/pdf/1602.00763.pdf, Last accessed: November 29, 2023.

[19] Deep Simple Online And Realtime Tracking, https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/, Last accessed: November 29, 2023.

[20] ByteTrack: Multi-Object Tracking by Associating Every Detection Box, https://arxiv.org/pdf/2110.06864.pdf, Last accessed: November 29, 2023.

[21] Evaluation Metrics for Multiple Object Tracking, https://arshren.medium.com/evaluation-metrics-for-multiple-object-tracking-7b26ef23ef5f, Last accessed: November 29, 2023.

[22] What is LLMs, https://datascientest.com/large-language-models-tout-savoir, Last accessed: November 29, 2023.

[23] What are LLMs, and how are they used in generative AI?, https://www.computerworld.com/article/3697649/what-are-large-language-models-and-how-are-they-used-in-generative-ai.html, Last accessed: November 29, 2023.

[24] Specialized LLMs: ChatGPT, LaMDA, Galactica, Codex, Sparrow,https://towardsdatascience.com/specialized-llms-chatgpt-lamda-galactica-codex-sparrow-and-more-ccccdd9f666f, Last accessed: November 29, 2023.

[25] What is BERT ? ,https://datascientest.com/bert-un-outil-de-traitement-du-langage-innovant, Last accessed: November 29, 2023.

[26] What is Llama ? ,https://ai.meta.com/blog/large-language-model-llama-meta-ai/, Last accessed: November 29, 2023.

[27] ChatGPT website,https://chat.openai.com/, Last accessed: November 29, 2023.

[28] What is RCNN ? , https://blog.roboflow.com/what-is-r-cnn/, Last accessed: November 29, 2023.

[29] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, https://arxiv.org/pdf/1506.01497.pdf, Last accessed: November 29, 2023.

[30] Understanding Single Shot Detector (SSD), https://medium.com/@elvenkim1/understanding-single-shot-detector-ssd-782e4b93aec3, Last accessed: November 29, 2023.

[31] You Only Look Once: Unified, Real-Time Object Detection, https://arxiv.org/pdf/1506.02640.pdf, Last accessed: November 29, 2023.

[32] Video Analytics SoftWare, https://sourceforge.net/software/video-analytics/, Last accessed: September 3, 2023.

[33] Software Project, https://www.tutorialspoint.com/software_engineering/software_project_management.htm, Last accessed: August 30, 2023.

[34] What is object tracking computer vision, https://blog.roboflow.com/what-is-object-tracking-computer-vision/#:~:text=Initially%2C%20the%20object%20tracking%20algorithm,until%20they%20leave%20the%20frame., Last accessed: September 14, 2023.

[35] Demo KiwiVision platform, https://www.youtube.com/watch?v=hThh_uRG-00, Last accessed: September 3, 2023.

[36] KiwiVision platform, https://www.genetec.com/fr/produits/securite-unifiee/omnicast/analyse-video, Last accessed: September 3, 2023.

[37] Frigate NVR platform, https://frigate.video/, Last accessed: September 3, 2023.

[38] Top 5 Objets Tracking Methods, https://medium.com/augmented-startups/top-5-object-tracking-methods-92f1643f8435, Last accessed: September 12, 2023.

[39] Metrics and Objets Tracking Algorithms, https://visailabs.com/evaluating-multiple-object-tracking-accuracy-and-performance-metrics-in-a-real-time-setting/, Last accessed: September 12, 2023.

[40] What is object detection computer vision, https://blog.roboflow.com/object-detection/, Last accessed: September 14, 2023.

[41] NLP and Transformers, https://towardsdatascience.com/transformers-89034557de14/, Last accessed: September 15, 2023.

[42] Amazon Rekognition, https://aws.amazon.com/fr/rekognition/, Last accessed: September 9, 2023.

[43] Google Cloud Video IA, https://cloud.google.com/video-intelligence?hl=fr, Last accessed: September 12, 2023.

[44] Benchmark performance YOLO versions, https://pytorch.org/hub/ultralytics_yolov5/, Last accessed: September 14, 2023.

[45] DeepSort vs ByteTrack, https://www.visobyte.com/2023/07/ByteTrack-vs-Deepsort.html, Last accessed: September 12, 2023.

[46] Object Tracking State, https://medium.com/@pedroazevedo6/object-tracking-state-of-the-art-2022-fe9457b77382, Last accessed: September 14, 2023.

[47] Microsft Video Indexer, https://vi.microsoft.com/en-us, Last accessed: September 13, 2023.

[48] Support IBM Waston Recognition, https://www.ibm.com/support/customer/csol/terms/?id=i128-0026&lc=en#detail-document, Last accessed: September 14, 2023.

[49] Platform tools for Computer Vision, https://roboflow.com, Last accessed: September 10, 2023.

[50] Open Computer Vision, https://opencv.org/, Last accessed: September 10, 2023.

[51] Django, the framework web for perfectionists, https://www.djangoproject.com/, Last accessed: September 10, 2023.

[52] Next JS, https://nextjs.org/, Last accessed: September 10, 2023.

[53] Docker, https://www.docker.com/, Last accessed: September 10, 2023.

[54] COCO Dataset, https://cocodataset.org/, Last accessed: September 11, 2023.

[55] NumPy Project, https://numpy.org/doc/stable/user/whatisnumpy.html, Last accessed: September 2, 2023.

[56] Matplotlib Project, https://matplotlib.org/, Last accessed: September 2, 2023.

[57] LangChain, https://python.langchain.com/, Last accessed: September 15, 2022.

[58] OpenAI API Library, https://platform.openai.com/docs/introduction, Last accessed: November 29, 2022.

[59] Background class in confuxion matrix , https://medium.com/@a0922/confusion-matrix-of-yolov8-97fd7ff0074e, Last accessed: November 29, 2022.

French summary

Mise en Contexte

La vidéosurveillance, un pilier crucial dans le domaine de la sécurité, joue un rôle déterminant dans la protection des biens et des personnes, tout en contribuant efficacement à la prévention de la criminalité. L'effet dissuasif des caméras de surveillance est notable, réduisant les actes criminels de manière significative, parfois jusqu'à 30% . Ces systèmes de surveillance sont devenus des outils incontournables pour la supervision des espaces publics et privés, apportant une sensation de sécurité accrue. Cependant, les systèmes de vidéosurveillance traditionnels sont confrontés à plusieurs défis, notamment leur dépendance à une surveillance humaine constante, qui peut entraîner des lacunes dans la couverture et un manque de réactivité en temps réel. De plus, la gestion des enregistrements vidéo peut s'avérer laborieuse et inefficace, ce qui conduit à une accumulation de données inexploitées et à un gaspillage de ressources. Ces limitations mettent en lumière la nécessité d'adopter des solutions plus avancées et automatisées pour améliorer l'efficacité de la sécurité.

Introduction de NextVision

Dans ce contexte, NextVision émerge comme une initiative ambitieuse visant à repenser la vidéosurveillance. L'objectif de NextVision est de transformer les systèmes traditionnels en une solution plus intelligente, flexible et conviviale. En intégrant des technologies d'intelligence artificielle (IA) de pointe, NextVision vise à automatiser la détection et l'analyse dans les systèmes de surveillance vidéo, permettant une surveillance plus efficace et ouvrant la voie à des interactions plus intuitives entre les utilisateurs et le système. L'ambition de NextVision est de créer un système où les données sont non seulement traitées et analysées de manière intelligente, mais fournissent également des informations pertinentes et actionnables, tout en réduisant la charge de travail des opérateurs humains et en optimisant l'utilisation des ressources.

Généralités sur l'Intelligence Artificielle et ses Applications

L'intelligence artificielle (IA) représente un tournant technologique majeur, offrant des capacités de résolution de problèmes complexes souvent au-delà des capacités humaines. Elle est particulièrement efficace pour des tâches répétitives et minutieuses. Grâce à sa capacité à traiter et analyser de vastes volumes de données, l'IA trouve des applications dans divers domaines, y compris la sécurité. Au cœur de l'IA se trouvent l'apprentissage automatique (Machine Learning, ML) et l'apprentissage profond (Deep Learning, DL). Le ML permet aux systèmes d'apprendre et de s'améliorer de manière autonome à partir des données collectées, sans être programmés explicitement pour chaque tâche.

Le DL, inspiré des mécanismes du cerveau humain, utilise des réseaux de neurones pour traiter les informations de manière complexe et stratifiée. Ces réseaux, en particulier les réseaux neuronaux convolutifs (CNN), sont cruciaux dans le domaine de la vision par ordinateur (Computer Vision, CV), permettant aux machines d'analyser et d'interpréter les images et les vidéos de manière similaire à la perception humaine. En parallèle, le traitement du langage naturel (Natural Language Processing, NLP) facilite des interactions plus fluides et naturelles entre les humains et les machines, avec des avancées comme les transformateurs et les grands modèles de langage, tels que ChatGPT, qui illustrent le potentiel du NLP dans la compréhension et le traitement des langues humaines.

En combinant la vision par ordinateur et le traitement du langage naturel, NextVision est capable de gérer et d'analyser des flux vidéo à partir de requêtes textuelles. Cette synergie permet à NextVision de fournir une surveillance plus efficace et interactive, améliorant significativement la sécurité et la réactivité du système.

Etat de l'Art de l'Existant et Avancées de NextVision

Dans le secteur de la vision par ordinateur, diverses solutions telles que KiwiVision, Frigate NVR, Amazon Rekognition, Microsoft Video Indexer et Google Cloud Video Intelligence ont été développées. Ces outils offrent des fonctionnalités variées, comme la transformation de vidéos en informations exploitables, la détection d'objets en temps réel, l'analyse de vidéos pour l'identification d'objets et de personnes, et l'extraction d'informations pertinentes à partir de vidéos.

Cependant, ces solutions présentent certaines limitations, notamment en termes de spécialisation en sécurité, d'exigences en expertise technique, de complexité des interfaces, de coûts, de personnalisation et d'évolutivité. Pour répondre à ces lacunes, NextVision a été développé comme une plateforme innovante alliant IA, vision par ordinateur et NLP. NextVision se distingue par sa flexibilité, son interface utilisateur intuitive, sa capacité de traitement en temps réel et son adaptabilité aux besoins spécifiques de la surveillance vidéo intelligente.

NextVision représente un progrès significatif dans le domaine de la surveillance vidéo intelligente, en intégrant les avancées en NLP et en vision par ordinateur. Cette solution offre une alternative plus adaptable aux outils existants, marquant un tournant dans la façon dont la surveillance vidéo peut être gérée et exploitée.

Technologies exploités

NextVision incarne une innovation dans la surveillance vidéo intelligente, intégrant des technologies avancées pour optimiser la sécurité et l'interaction utilisateur. Au cœur de ce système réside un module de traitement du langage naturel , permettant aux utilisateurs de formuler des requêtes textuelles en temps réel qui sont immédiatement analysées et converties en commandes exécutables.

Une caractéristique distinctive de notre est son utilisation de YOLO v8 pour la détection d'objets. Ce modèle, reconnu pour sa rapidité et sa précision, a été entraîné sur un jeu de données spécifiquement conçu pour des applications de sécurité, ciblant des objets tels que des armes à feu, des individus, des incendies, des cigarettes et des téléphones.

Concernant le cadre de développement web, NextVision utilise Django et Next.js. Django, un framework Python hautement versatile, est utilisé pour la construction robuste du backend. Il se distingue par sa sécurité, sa scalabilité et sa capacité à gérer des applications complexes. Next.js, une bibliothèque de React, est employé pour le frontend, offrant des fonctionnalités telles que le rendu

côté serveur et la génération de sites statiques, ce qui permet une expérience utilisateur fluide et réactive.

La communication en temps réel entre le client et le serveur est gérée grâce à la technologie WebSocket. Contrairement au modèle de communication HTTP classique, WebSocket permet une interaction bidirectionnelle continue, essentielle pour la surveillance en temps réel et la réactivité du système.

En ce qui concerne l'aspect NLU (Natural Language Understanding), NextVision intègre l'API ChatGPT avec Langchain. Langchain est un cadre de traitement du langage qui facilite l'intégration de modèles de langage comme GPT-3 dans des applications diverses. GPT-3, avec son préentraînement sur une vaste gamme de textes, offre une compréhension et une génération de langage naturel remarquables, rendant les interactions utilisateur plus intuitives et naturelles.

NextVision repousse les limites de la surveillance vidéo grâce à son architecture innovante et à l'intégration de technologies de pointe telles que Django, Next.js, WebSocket, ainsi que le traitement avancé du langage naturel via GPT-3 et Langchain. Cette combinaison offre une plateforme de surveillance sophistiquée, sécurisée et hautement interactive.

Notre Approche avec NextVision

NextVision est un système de vidéosurveillance innovant qui combine le traitement du langage naturel (NLP) et la détection d'objets pour offrir une solution de surveillance vidéo plus intelligente et interactive. Notre système se distingue par deux modules principaux : un module de traitement du langage et un module de détection d'objets avancé.

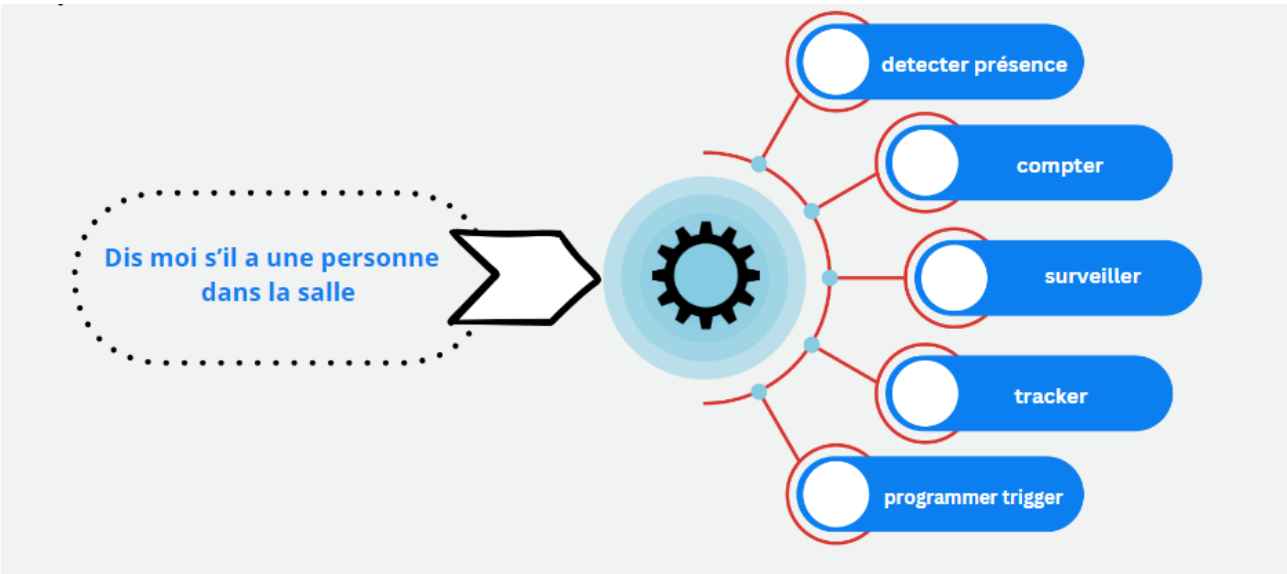

Figure 3.21: NEXTVISION Understanding request

Le module de NLP, exploitant GPT-3 et Langchain, est conçu pour interpréter les requêtes textuelles des utilisateurs et les convertir en commandes actionnables (3.21). Ces commandes peuvent couvrir une gamme d'actions, telles que la détection de présence, le comptage, la surveillance générale, et la programmation de déclencheurs basés sur des événements spécifiques. Cette fonctionnalité rend l'interaction avec NextVision naturelle et efficace, en permettant aux utilisateurs de communiquer directement avec le système dans un langage quotidien.

En termes de détection d'objets, NextVision utilise YOLO v8 pour développer un modèle pré-

entraîné, spécifiquement optimisé pour identifier cinq types d'objets : pistolets, personnes, cigarettes, téléphones et couteaux. Cette précision dans la reconnaissance d'objets est essentielle pour répondre efficacement aux exigences de sécurité et de surveillance. La collecte et l'annotation des données nécessaires pour le modèle de détection d'objets ont été réalisées via Roboflow, permettant de construire un ensemble de données sur mesure et améliorant la capacité de YOLOv8 à détecter les objets mentionnés dans divers contextes de surveillance.

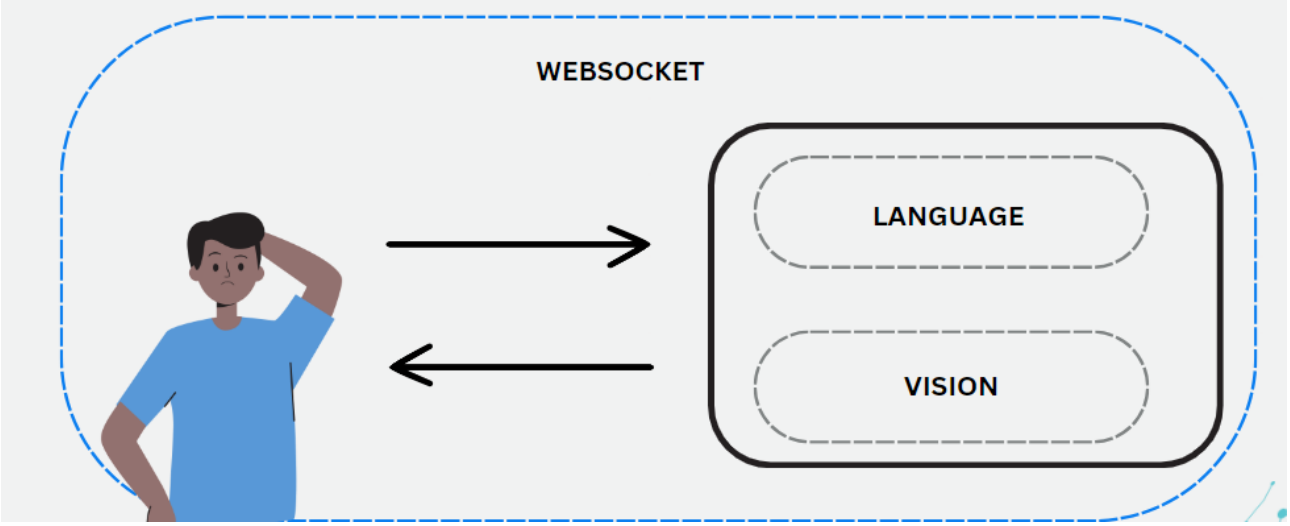

Figure 3.22: NEXTVISION system summary

Pour une intégration harmonieuse de ces deux modules, nous avons utilisé un système de WebSockets (3.22). Cette technologie assure une communication fluide et en temps réel entre le module de NLP et le module de détection d'objets, garantissant ainsi une réactivité et une coordination optimales du système NextVision dans son ensemble.

Résultats Obtenus et Limitations

Le projet NextVision, comme abordé dans les chapitres précédents, a combiné un état de l'art poussé avec l'utilisation de méthodes et outils spécifiques pour aboutir à un modèle efficace. Ce chapitre est dédié à la présentation des résultats obtenus.

La solution NextVision s'articule autour de trois composants clés : un modèle de vision pour la détection et le suivi d'objets, la conteneurisation du modèle via Docker, et le développement d'une API englobant toute la logique du système, y compris l'authentification et la gestion des flux vidéo en temps réel. Pour compléter cela, une plateforme web interactive a été créée, permettant aux utilisateurs d'interagir facilement avec l'API.

Le développement du module de vision a impliqué la création d'un système capable de traiter des images et de détecter des objets en utilisant la technologie YOLO. Ce modèle a été spécialement adapté pour reconnaître cinq classes cibles. Roboflow a joué un rôle crucial dans la collecte et l'annotation des données, facilitant le développement du module. Le processus a inclus la préparation des données, l'augmentation des données pour enrichir le jeu de données, et l'entraînement du modèle avec YOLOv8, utilisant Google Colab Pro pour une puissance de calcul significative.

Les performances du modèle ont été évaluées rigoureusement, en utilisant des méthodes comme la matrice de confusion pour mesurer l'exactitude de la classification. Les graphiques de résultats et les courbes de précision-rappel ont été analysés pour évaluer l'efficacité du modèle. Des tests sur

des lots d'images ont montré que le modèle identifiait correctement les objets ciblés, malgré certains défis, notamment dans la détection des couteaux.

En plus du module de vision, une API a été développée avec des points de terminaison spécifiques pour la gestion des sources vidéo et l'authentification. Des tests ont été effectués pour s'assurer de la fonctionnalité de l'API, notamment en utilisant des WebSocket pour des interactions en temps réel.

L'interface utilisateur a été soigneusement conçue pour offrir une expérience utilisateur fluide, avec des pages dédiées à l'inscription, à la connexion, au tableau de bord, et à une "aire de jeu" où les utilisateurs peuvent interagir directement avec le système de surveillance vidéo. La performance globale du système a été évaluée en tenant compte de divers facteurs techniques, et bien que le modèle pré-entraîné ait présenté certaines limites, les résultats étaient globalement satisfaisants.

En conclusion, bien que NextVision ait réussi à atteindre plusieurs de ses objectifs, l'évaluation a révélé des imperfections et des faiblesses, notamment en ce qui concerne la base de connaissances pour la formation. Ces domaines nécessitent une attention et une amélioration futures pour renforcer les capacités et la performance de l'application.

Conclusion et Perspectives

NextVision associe vision par ordinateur et traitement du langage naturel pour offrir une expérience utilisateur exceptionnelle. Il représente une avancée dans le domaine de la surveillance vidéo intelligente, ouvrant de nouvelles possibilités pour une sécurité renforcée et un suivi efficace. Cependant, il existe des opportunités passionnantes pour des travaux futurs. Tout au long de ce mémoire, nous avons relevé le défi de faire comprendre et interpréter son environnement à une machine. Nous avons adopté une approche basée sur des règles pour notre modèle de détection d'objets et de traitement du langage naturel. Bien que cette approche ait fourni des perspectives , notre modèle n'est pas encore optimal. Pour améliorer l'outil à l'avenir, nous envisageons plusieurs idées . Une option est de rassembler et d'étiqueter plus de données, ce qui permettrait à notre modèle de reconnaître une gamme plus large d'objets. Une autre approche à considérer est la Légende Visuelle, impliquant la fourniture de descriptions de scènes plus détaillées. Nous pouvons également explorer l'utilisation d'outils de vision par ordinateur avancés pour construire des fonctionnalités plus puissantes. Améliorer la capacité de notre algorithme à comprendre ce qui est dans une image est crucial. De plus, nous pouvons utiliser des actions déclenchées par des événements basées sur des conditions spécifiques à travers des files d'attente pour améliorer la performance de la sécurité.

Contents