People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of May 8, 1945 - Guelma

Faculty of Mathematics, Computer Science and Material Sciences

Department of Computer Science

Master's Thesis

Speciality: Computer Science

Option: Information Systems

Theme

Leveraging AI and IoT for Enhanced Surveillance and Efficient Management of Water Dams

---

Jury Members:

Chairwoman: Dr. Asma Boudria

Supervisor: Dr. Khaled Halimi

Examiner: Pr. Brahim Farou

POLE PRO Representative: Dr. Lazhar Hani

Presented by:

Yassamine Boumelit

June 2025

Acknowledgements

At the beginning of this journey, I could not have imagined where the path would lead. Sometimes, life takes us in directions we had not planned, yet through those unexpected roads, we grow, adapt, and discover new strengths within ourselves. And for that, I am grateful.

Above all, I thank **Allah**, the Most Merciful, the Most Wise, for guiding me through this chapter of my life. His wisdom placed me on this path, and His mercy carried me through each challenge. *Alhamdulillah* for the strength, the patience, and the faith that allowed me to complete this work.

I extend my sincere and heartfelt gratitude to my supervisor, **Dr. Halimi Khaled**, for his unwavering support, thoughtful guidance, and constant encouragement throughout this process. I am truly honored to have worked under his supervision, and I will always value the lessons learned under his mentorship.

To my **mother**, your endless love and quiet prayers were my greatest source of strength.
To my **father**, your wisdom and support have shaped who I am.
And to my **brother Islam**, thank you for your presence, your encouragement, and your pride in every small success I made. You are my foundation, and I am forever grateful.

To my dear **friends**, thank you for being a comforting presence throughout this journey—for the shared moments of doubt and joy, for your words of encouragement, and for making the hardest days easier to bear.

And to my **fellow batchmates**, we walked this path together—with its trials, late nights, and shared goals. Our time together will always be a treasured part of this experience. I wish you all continued success, fulfillment, and strength as you move forward into the future.

This work is a product of effort, faith, and the support of those who believed in me. It is with deep gratitude that I close this chapter, carrying with me the lessons, memories, and people who made it all possible.

Boumelit Yassamine

Résumé

Les barrages font face à des menaces multiples pouvant compromettre leur sécurité et mettre en péril des vies humaines, telles que l'accumulation d'objets flottants, les situations de noyade non détectées, et l'apparition progressive de fissures structurelles. Ces problématiques soulignent le besoin urgent d'un système de surveillance intelligent, capable de réagir rapidement et de manière autonome.

Pour répondre à ces défis, un système de détection basé sur l'intelligence artificielle et la vision par ordinateur a été conçu afin d'identifier automatiquement trois types de risques critiques : les objets flottants, les noyades et les fissures dans les structures hydrauliques.

Le système repose sur l'architecture YOLOv8 en deep learning pour réaliser des détections en temps réel à partir d'images issues de caméras de surveillance. Chaque module a été entraîné sur des jeux de données annotés, avec un traitement rigoureux incluant le prétraitement, l'augmentation de données et l'ajustement des hyperparamètres.

Les résultats obtenus dans un environnement contrôlé montrent des performances prometteuses pour les trois modules de détection. Cette solution constitue une avancée vers une plateforme intelligente et adaptable pour améliorer la gestion et la sécurité des barrages en Algérie.

Mots-clés : sécurité des barrages, surveillance intelligente, YOLOv8, deep learning, vision par ordinateur, détection d'objets flottants, noyade, fissures

Abstract

Ensuring dam safety requires proactive monitoring of multiple critical risks, including floating debris, potential drowning incidents, and the gradual emergence of structural cracks. In response to these challenges, this work presents an integrated intelligent system based on computer vision and artificial intelligence, designed to detect and classify these hazards in real-time from visual data.

The proposed system unifies three complementary detection modules within a single platform: floating object detection, drowning detection, and crack detection. Each module is built upon the YOLOv8 deep learning architecture and trained on dedicated, annotated datasets with tailored preprocessing and optimization techniques.

This unified approach enables the system to automatically analyze surveillance imagery, generate accurate alerts, and support decision-making in dam management operations. The experimental results, obtained in a controlled environment, confirm the system's effectiveness and robustness across all three detection tasks. This work lays the foundation for a future deployable solution to enhance dam safety through intelligent and autonomous monitoring.

Keywords: Integrated intelligent system, dam safety, deep learning, YOLOv8, computer vision, floating object detection, drowning detection, crack detection, real-time monitoring.

الخلاصة

واجهت المنشآت المائية الكبرى، وعلى رأسها السدود، تحديات متزايدة تتعلق بالمراقبة الدورية والسلامة الهيكلية. من أبرز هذه التحديات: تراكم الأجسام الطافية التي قد تعيق تشغيل المنافذ، خطر حوادث الغرق بسبب غياب الإنذار المبكر، وظهور تشققات في الهياكل الخرسانية التي قد تؤدي إلى أضرار جسيمة إذا لم تُكتشف في الوقت المناسب. غالبًا ما تعتمد مراقبة هذه المخاطر على المعاينة اليدوية أو الكاميرات التقليدية، مما يجعل الاستجابة بطيئة وغير فعالة، خاصة في الحالات الحرجة أو في المناطق المعزولة.

تكتسب هذه الإشكالية أهمية متزايدة في ظل التغيرات المناخية وارتفاع الاعتماد على السدود لتخزين المياه وتوليد الطاقة، ما يستدعي حلولًا ذكية لضمان التدخل السريع وتقليل الخسائر. إن تحسين آليات المراقبة لا يمثل فقط خطوة نحو الحد من المخاطر، بل يعزز أيضًا من كفاءة التشغيل ويقلل من تكاليف الصيانة الطويلة الأمد.

في هذا السياق، يقترح هذا المشروع تطوير نظام ذكي يعتمد على تقنيات الذكاء الاصطناعي، وبالخصوص خوارزميات الرؤية الحاسوبية والتعلم العميق، لرصد ثلاث حالات حرجة داخل بيئة السدود: الأجسام الطافية، حالات الغرق، والتشققات في الهياكل. وقد تم تدريب هذا النظام باستخدام نماذج حديثة مثل YOLOv8 وغيرها، قصد تحقيق أداء عالٍ من حيث الدقة والسرعة، ما يجعله حلاً عمليًا يعزز من فعالية المراقبة الوقائية في السدود.

الكلمات المفتاحية: السدود، الذكاء الاصطناعي، الرؤية الحاسوبية، التعلم العميق، الكشف عن الأجسام الطافية، كشف الغرق، كشف التشققات، YOLOv8، السلامة الهيكلية، أنظمة المراقبة الذكية.

Contents

ListofFigures

ListofTables

General introduction

Dams are essential infrastructure for water management, energy production, and environmental balance. However, ensuring their operational safety remains a complex and high-risk challenge. Dams are vulnerable to various hazards, including the accumulation of floating debris, accidental sinkings, and the formation of structural cracks, all of which can jeopardize their stability and safety. Traditional monitoring approaches rely heavily on manual inspections and fixed surveillance cameras, which are labor-intensive, reactive, and susceptible to human error. Recent advances in artificial intelligence (AI), particularly in the fields of deep learning and computer vision, have opened new avenues for improving anomaly detection and automating dam monitoring. In this context, our thesis proposes an intelligent system capable of detecting and classifying three types of safety-related anomalies in dam environments: floating objects, sinking accidents, and structural cracks. Our approach is based on developing and training object detection models using carefully selected, pre-processed datasets. These models were evaluated under various environmental conditions to assess their accuracy and reliability in realistic scenarios. To support demonstration and testing, we developed a basic programming interface that displays the model functionality in a simplified monitoring environment.

This thesis is divided into four chapters:

Chapter 1: Technological Trends in Dam Safety - Artificial Intelligence, the Internet of Things, and Hazard Detection Chapter 1 provides a comprehensive background on the general framework of the research problem, highlighting the reality of dams in Algeria, their strategic importance in securing water resources, and the challenges they face in maintenance and monitoring. The chapter begins with an overview of Algerian dams, followed by a presentation of their economic and social importance, with examples of the country's largest dams. It then moves on to discuss the challenges associated with managing these vital facilities, particularly with regard to monitoring floating objects, sinkings, and structural cracks, which pose a direct threat to public safety and dam operational efficiency. The chapter also explores the need to employ smart systems powered by the Internet of Things (IoT) and modern artificial intelligence technologies to achieve proactive and effective monitoring. It elaborates on the mechanism of data transmission and storage via IoT networks, comparing the capabilities of human and machine learning in handling visual data. The chapter concludes with an introductory overview of the three main technical areas addressed by the project: floating object detection, crack detection, and sinking detection. This paves the way for subsequent chapters to detail the proposed technical solutions. Chapter 02: Deep Learning Approaches to Object Detection this chapter addresses the basic technical aspects related to deep learning-based computer vision algorithms, focusing on their use in object detection tasks in images. The chapter opens with a general introduction that paves the way for an in-depth explanation of the most important algorithms used in this field. It reviews Faster R-CNN, RetinaNet, and YOLO in its various versions, from YOLOv2 to YOLOv8. The chapter then moves on to discuss previous related work, reviewing the most prominent studies on detecting floating objects and sinking cases in aquatic environments using artificial intelligence techniques, as well as research related to detecting structural cracks in dams. Through this presentation, the chapter provides an analytical overview of previously adopted approaches.

Chapter 03: Conception Chapter Three focuses on the design and practical implementation of the proposed intelligent system, detailing the technical procedures used to prepare the data and train artificial intelligence models for each of the three tasks: detecting floating objects, detecting sinking cases, and monitoring structural cracks. The chapter begins with a general introduction, then moves on to a systematic presentation of the preprocessing steps, starting with collecting data from various sources, through cleaning and segmenting, and finally addressing the challenges associated with class diversity and imbalanced distribution. The chapter also explains data augmentation techniques used to improve model performance in cases of few classes.Separate sections are devoted to training each model, explaining the chosen version of the YOLO algorithm and the reasons for its selection, in addition to justifying the selection of appropriate hyperparameters for each case. It also details how the datasets were combined and balanced, especially in cases of class imbalance. The chapter also addresses the technical challenges faced in the training process, such as limited computational resources or image complexity. It concludes with an analysis of the YOLOv8 architecture and explains why this algorithm was adopted as the final choice for the proposed model.

Chapter04: Implementation presents the practical aspect of the project, evaluating the performance of the trained models and analyzing their results in a real-world operating environment. It begins by defining the development environment used, describing the hardware and software characteristics used during the training and testing processes. It then goes on to detail the training and validation processes, presenting graphs and statistical measures of the models' performance in detecting floating objects, cracks, and sinking situations. The chapter also includes an analysis of the test results, reviewing the model accuracy and other performance indicators such as reliability and efficiency, and providing an in-depth discussion of the results to understand the strengths and shortcomings of the proposed system. The chapter also devotes a section to presenting the user interface developed for the system's demonstration, highlighting how images can be manually entered and real-time predictions obtained. The chapter concludes with an evaluative conclusion summarizing the most important findings and opening up prospects for future system improvement within the framework of more comprehensive field applications.

Chapter 1

Technological Trends in Dam Safety: AI, IoT, and Risk Detection

1.1 Introduction

Dams are among the most important means by which we conserve and exploit water and hydropower, which are used in various fields, such as agriculture and providing drinking water. They also play a fundamental role in protecting against natural disasters. This engineering achievement is considered one of the most important vital infrastructures. However, despite all its benefits, it also presents challenges. With prolonged use, concrete dams become susceptible to corrosion, cracking, and disintegration, weakening their structural integrity and reducing their ability to withstand pressure. This can lead to disasters like the one that occurred in Libya.

The presence of foreign objects within dams also poses a significant threat to the infrastructure of these vital facilities and the surrounding environment. They can cause blockages in pumps and turbines used for generating power or pumping water, damaging equipment and disrupting vital dam operations. The fall of dead animals and waste can also contaminate water, threatening human health and living organisms. Therefore, early detection of objects in dams and potential concrete problems, such as cracks, can save us from disasters and costly maintenance costs.

Dams are tourist destinations where visitors enjoy the natural beauty and water activities. However, the presence of deep water makes them dangerous areas, especially for children who may be exposed to drowning accidents. This is where smart drowning detection systems come in. They monitor the water and alert the relevant authorities immediately after an emergency occurs.

This is what prompted us to develop a smart monitoring system for water dams, using Internet of Things sensors and artificial intelligence algorithms, to ensure safety and optimal management of water resources, making dams safer recreational areas for families and visitors.

1.2 Economic and environmental importance of dams

1.2.1 Economic Role of Dams

- **Agriculture:** Dams ensure a stable water supply for irrigation, which increases crop productivity and supports food security. Globally, agriculture accounts for more than 69% of freshwater withdrawal [1, 2].

- **Hydropower:** Hydroelectric power generated by dams offers a renewable and cost-effective energy source. For instance, Turkey has a hydropower potential of 433 GWh/year, making it a strategic asset for national energy planning [3].

- **Flood control:** By regulating river flow, dams help protect downstream residential and agricultural areas. A study in China's Wangmo Basin showed that dam implementation reduced flooded housing areas by 12.9% to 30.2% [4].

- **Tourism and local economy:** Dams and their surrounding artificial lakes create opportunities for ecotourism, recreation, and job creation. The Tehri Dam in India, for example, attracts tourists for activities like boating and medical tourism [5].

1.2.2 Environmental role of dams

Dams also provide key environmental benefits, including:

- **Biodiversity support:** Reservoirs create aquatic habitats that foster fish, birds, and other species [6].
- **Saltwater intrusion control:** In coastal areas, dams help block seawater from contaminating groundwater sources [6].
- **Lower carbon emissions:** Hydropower reduces dependence on fossil fuels, contributing to climate change mitigation [6].
- **Soil and land protection:** By regulating water flow, dams prevent soil erosion and limit desertification [6].

1.3 Dams in Algeria

1.3.1 Overview of Dams in Algeria and their importance

Algeria currently has 81 dams with a total capacity exceeding 9 billion m^3. By 2030, the number is expected to reach 120, storing up to 12.5 billion m^3. These dams support agriculture, drinking water supply, hydroelectric power, and flood control [7].

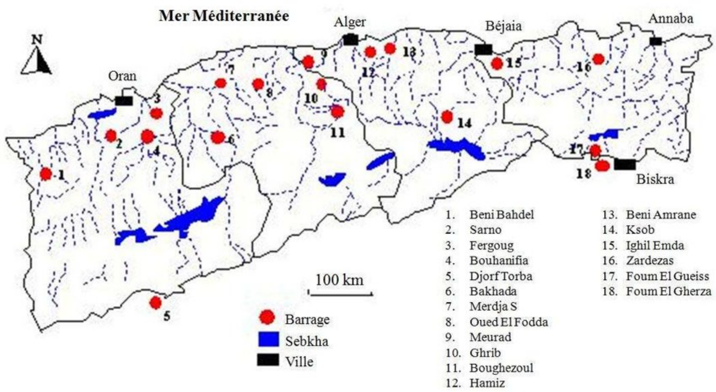


FIGURE 1.1: Geographical location of dams in Algeria [8]

Dams are a key component of Algeria's national water and economic strategy, playing a vital role in:

- **Industrial support:** Providing water to major industrial zones and manufacturing operations across the country [9].
- **Agricultural irrigation:** Enhancing food security by ensuring consistent water supply for crops, especially in drought-prone regions. For example, Beni Haroun Dam irrigates six key agricultural provinces [9].
- **Drinking water supply:** Ensuring access to potable water for a wide range of communities and cities [9].

1.3.2 Examples of the largest dams in Algeria

- **Beni Haroun Dam (Mila):** Stores 228 million m^3 annually. It supplies drinking water, irrigates 30,000 hectares, and transfers water to six wilayas [10].

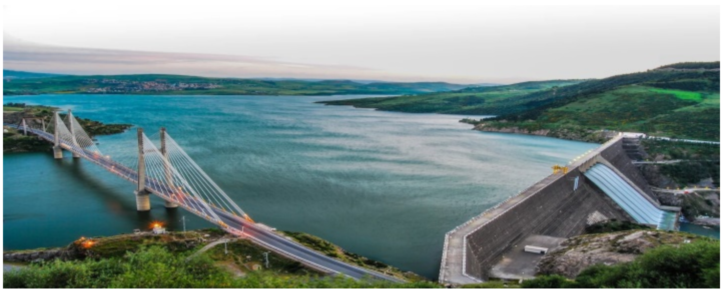


FIGURE 1.2: Beni Haroun Dam [11]

- **Koudiat Acerdoune Dam (Bouira):** With a storage of 640 million m^3, it provides water for 1.58 million people and irrigates 19,000 hectares [10].

FIGURE 1.3: Koudiat Acerdoune Dam [11]

- **Taksebt Dam (Tizi-Ouzou):** Holds 175 million m^3 and supplies both drinking and industrial water to the region [10].

FIGURE 1.4: Taksebt Dam [11]

1.3.3 Dams Management Challenges in Algeria

Dams in Algeria face various environmental and operational challenges affecting their performance and sustainability:

- **Siltation and sedimentation:** Sediment accumulation reduces storage capacity and affects water quality, especially in semi-arid regions like Algeria, where erosion and surface runoff are common challenges [10]. Globally, over 25% of dam storage capacity is at risk of being lost by 2050 [12, 13].

- **Floating debris and waste:** Objects such as wood, plastic, and industrial waste can block intake systems and damage mechanical components, increasing maintenance costs and flood risks due to obstructed discharge channels [14, 15].

- **Water pollution:** Industrial, urban, and agricultural waste contributes to poor water quality in several dams (e.g., Ouled Malouk), requiring advanced treatment before use [15].
- **Erosion and mechanical wear:** Coarse sediments can erode gates and pipelines, reducing dam lifespan and demanding costly repairs [15].
- **Drowning incidents:** In 2023 alone, 25 people drowned in Algerian dams, with 96% of cases occurring in unmonitored zones. This highlights the need for intelligent safety systems and public awareness [16, 17].
- **Toxic algal blooms:** Dams like Hammam Debagh have experienced cyanobacterial blooms (e.g., *Planktothrix rubescens*), which produce toxins harmful to human health and disrupt water treatment processes. Early detection systems are essential to mitigate these risks [14].

1.4 The Need for Intelligent Systems to Detect Floating Objects, Drowning People, and Structural Cracks

1.4.1 Threats Posed by Floating Objects, Cracks, and Drowning in Dams

1) Floating Objects

Floating debris such as wood, plastic, and waste poses operational and environmental challenges for dams. It can block intakes, reduce turbine efficiency, damage mechanical parts, and increase maintenance costs [15, 18, 19]. Additionally, decomposing waste deteriorates water quality and promotes bacterial growth, raising health concerns [20]. These impacts highlight the importance of early detection systems and preventive cleaning strategies.

2) Cracks in Dams

Cracks threaten dam stability by allowing water infiltration, weakening structural integrity, and increasing collapse risk. Their presence in seismic zones exacerbates danger due to fluctuating internal pressures during earthquakes [21]. Effective monitoring and timely intervention are essential to prevent structural failure.

3) Drowning Incidents

Low-head dams often conceal hydraulic hazards such as submerged hydraulic jumps, which can trap individuals in deadly recirculating currents. Despite their small size, these dams contribute to dozens of fatalities annually worldwide [22]. Intelligent surveillance systems and hydrodynamic risk assessment are necessary to reduce such accidents.

1.4.2 Traditional Systems vs Intelligent Systems

Traditional dam monitoring methods—such as visual inspections, manual measurements, and basic statistical models—have long been used to detect cracks, floating debris, and drowning incidents. However, these methods face major limitations: they are labor-intensive, lack real-time data, and rely heavily on human intervention [23, 24].

1) Traditional Dam Monitoring Systems

- **Visual Inspection and Manual Instrumentation:** These techniques depend on human observation and mechanical tools like pressure gauges. They are slow, error-prone, and unable to detect hidden or sudden structural changes [23, 24].
- **Statistical Models:** Traditional models (e.g., HST, MLR) rely on historical data and assume gradual degradation, making them ineffective for predicting sudden failures [24].
- **Drowning Detection:** Despite the presence of lifeguards, drowning incidents remain frequent due to fatigue, distractions, and lack of training [25].

- **General Limitations:**
  - No real-time monitoring or early warning systems.
  - High dependency on human input increases error margins.
  - Inability to process large or irregular data sets [23, 24].
  - Lack of predictive capability for future risks [23].

These limitations highlight the need for intelligent systems that integrate AI, IoT, and computer vision to provide automated, real-time, and predictive monitoring of dams.

2) Intelligent Dam Monitoring Systems

To overcome the limitations of traditional methods, intelligent systems based on Artificial Intelligence (AI), the Internet of Things (IoT), and Computer Vision have transformed dam monitoring by offering real-time, automated, and accurate detection capabilities.

- **Remote Sensing and IoT Integration:**
  - **Structural Monitoring:** Geodetic tools like TLS and SAR provide accurate deformation measurements. Combined with IoT, they enable continuous monitoring and early warning for structural risks [25].
  - **Smart Sensor Networks:** IoT sensors collect real-time data (e.g., pressure, temperature, seepage) and transmit it via Wi-Fi, LoRaWAN, or 4G/5G to the cloud. AI then analyzes this data to detect anomalies, allowing officials to respond quickly via mobile applications [26].
- **AI and Computer Vision:** Advanced models such as SSD, Faster-RCNN, and YOLOv5 are used to detect floating debris and drowning people in surveillance images and videos. These models offer high accuracy and speed, making them suitable for real-time water monitoring systems [27].

1.5 Role of the Internet of Things (IoT)

1.5.1 Definition of Internet of Things

is a network of connected physical objects equipped with sensors and software to collect and exchange data. It links the physical and digital worlds to improve efficiency and control [28].

1.5.2 The role of the Internet of Things in the field of dams

IoT improves dam management by using sensors to monitor water levels, pressure, temperature, and environmental changes. This enables real-time detection of floods, structural issues, and risks to dam stability [29, 30]. IoT enables real-time data transmission to the cloud, where AI analyzes patterns to predict risks. Motion sensors, for example, help detect floating objects and potential drowning cases [31].

1.5.3 Data Collection Using the Internet of Things

Types of Sensors Used in Dams

IoT technologies utilize various sensors to monitor dam conditions and detect potential threats. These include:

1. **Motion & Object Detection Sensors:**
    - **LiDAR (Light Detection and Ranging):** Uses laser pulses to measure distances and detect surface changes such as landslides or floating debris. Example: LeddarTech LeddarVu8 [32].
    - **Radar (Radio Detection and Ranging):** Emits electromagnetic waves to detect moving objects and environmental patterns with high accuracy. Example: Vaisala's radar systems [33].
2. **Water Level Sensors:**

- **Ultrasonic technology:** Measures the distance to the water surface by analyzing reflected sound waves [34].
- **Radar-based level sensors:** Use electromagnetic pulses to track water levels even in harsh conditions. Example: Endress+Hauser Micropilot FMR6x [35].

3. **Temperature & Humidity Sensors:**

   - **Infrared (IR) Sensors:** Detect temperature from a distance without physical contact using infrared radiation. Example: Fluke Ti480 PRO [36].
   - **Humidity Sensors:** Monitor moisture levels in the air or materials surrounding the dam to assess risk factors [37].

4. **Pressure Sensors:** Measure internal water pressure and structural stress to detect early signs of cracks or erosion [38].

Transmission Mechanism and Data Storage via IoT

- **Transmission Mechanism:**
  - **LoRaWAN:** A long-range, low-power wireless protocol for transmitting small data packets from sensors [39].
  - **Zigbee:** A short-range, low-energy protocol using mesh networking, ideal for connecting multiple sensors reliably. Operates at 2.4 GHz [40].
- **Data Storage:**
  - **Cloud storage:** Allows scalable and remote access to collected data, managed by third parties [40].
  - **Edge computing:** Processes data locally near the source, enabling real-time reactions to anomalies [41].
  - **Local storage:** Stores data on-site using physical devices, offering more control and independence [42].

1.6 Concepts Related to AI and ML and DL

1.6.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to systems capable of perceiving their environment and making decisions to achieve goals [43, 44]. Technically, IEEE defines AI as data processing, learning from experience, and autonomous decision-making, while ISO emphasizes pattern analysis and knowledge extraction [45]. These definitions highlight AI's evolution as a field focused on developing adaptive, intelligent systems that can learn, reason, and interact independently across sectors like healthcare, finance, and transport [46].

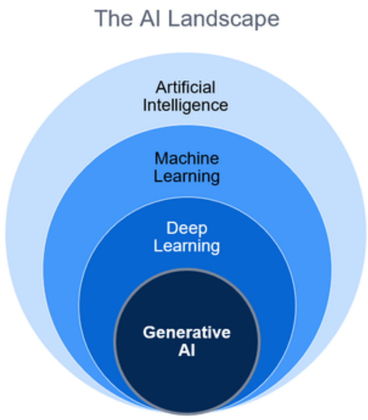


FIGURE 1.5: Different subdomain AI, ML, ANN and DL [47]

1.6.2 Machine Learning

Machine learning (ML), a branch of AI, enables systems to learn from data and make decisions without explicit programming. It includes supervised, unsupervised, semi-supervised, and reinforcement learning. ML models use statistical methods to classify data, predict outcomes, and detect patterns. Despite challenges like the need for strong math skills, quality data, and algorithm expertise, ML greatly improves efficiency and workflow optimization. [46].

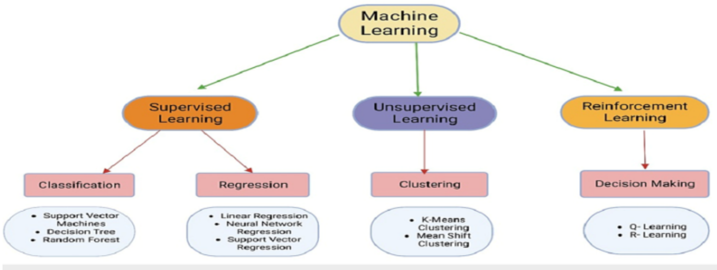


FIGURE 1.6: Machine learning types

[48]

Artificial Intelligence VS Machine Learning

Artificial Intelligence (AI)	**Machine Learning (ML)**
A computer science field that builds systems simulating human intelligence (learning, reasoning, problem-solving).	A subset of AI that develops algorithms to learn from data and improve without explicit programming
Encompasses technologies like ML, NLP, and robotics.	Focuses on data-driven models and algorithms.
Developing systems that can perform tasks that require human intelligence.	Enable systems to learn from data and make decisions or predictions based on that.
It may operate using pre-programmed rules and knowledge or by learning from data.	It relies entirely on data to learn and extract patterns.

TABLE 1.1: Differences between Artificial Intelligence and Machine Learning [49, 50]

Human learning VS Machine learning

Machine learning	**Human learning**
Analyzes large datasets using algorithms.	Learns through experience and interaction.
Needs retraining to adapt to new data or tasks.	Learns and adapts to new situations
Pattern-based, lacks awareness.	Understands, reasons, and decides contextually.
Needs data and training.	Learns independently via curiosity and exploration.
Struggles with noisy/incomplete data without preprocessing.	Analyzes incomplete info and decides wisely.

TABLE 1.2: Differences between Human learning and Machine learning [51]

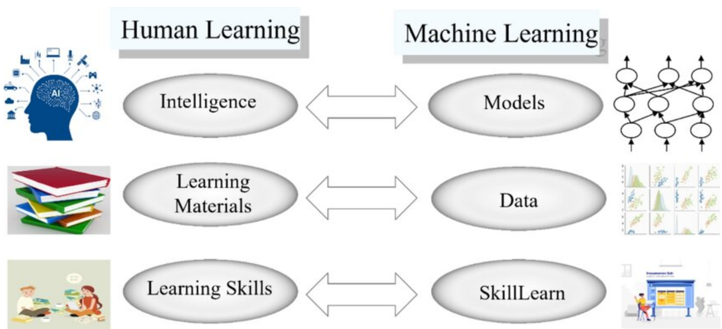


FIGURE 1.7: Human learning VS Machine learning

[52]

1.6.3 Deep Learning

Deep learning, a subset of AI, mimics the human brain to recognize complex patterns in images, text, and audio—improving analysis, prediction, and automation of tasks like image recognition and speech-to-text conversion[53].

Deep Learning Subfields

- **Artificial Neural Networks (ANNs):**Brain-inspired models composed of interconnected layers (input, hidden, output) that process data, recognize patterns, and support decision-making [54].

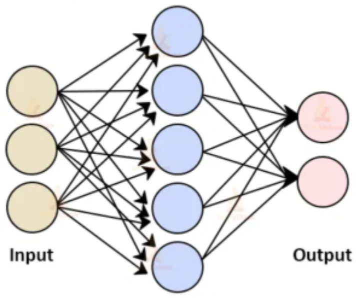


Figure 1.8: Architecture of ANN

[55]

- **Convolutional Neural Networks (CNNs):** A specialized type of ANN for image recognition, integrating feature extraction within the network. CNNs are efficient for processing image data and widely used in computer vision tasks like YOLO, Faster R-CNN, and SSD [54].

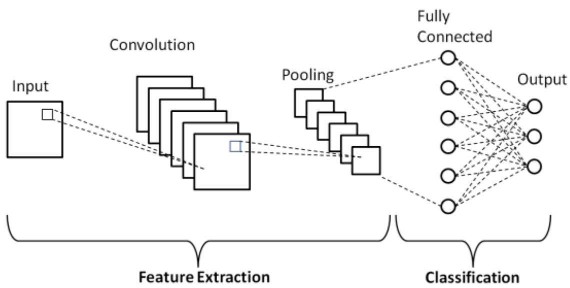


Figure 1.9: Architecture of CNNs

[56]

- **Deep Neural Networks (DNNs):** Consist of an input layer, multiple hidden layers, and an output layer. They use weights and activation functions to transform inputs and can approximate any function, making them useful in vision, speech, fraud detection, and predictive maintenance [57].

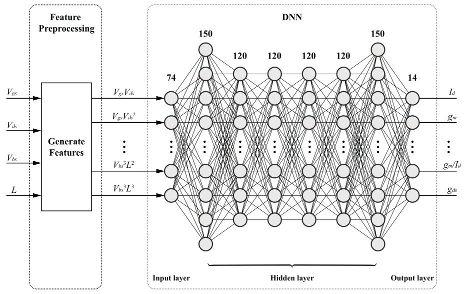


Figure 1.10: Architecture of DNNs

[58]

- **Recurrent Neural Networks (RNNs):** RNNs are neural networks designed for sequential data. They use recurrent connections to retain previous inputs, making them suitable for tasks like speech recognition, translation, and time series analysis[59].

- **Long Short-Term Memory Networks(LSTM):**LSTMs are a type of RNN designed to retain information over long sequences. They address the vanishing gradient problem, making them suitable for tasks requiring long-term memory in deep learning. [60].

- **Generative Adversarial Network (GAN):** GANs consist of a generator and a discriminator competing to create realistic synthetic data. They are widely used in image generation, content creation, and data augmentation [61].

- **Reinforcement Learning (RL):** A type of AI where an agent learns by interacting with its environment to maximize rewards, adjusting its actions based on feedback through trial and error[62].

- **Transfer Learning:**A technique where a model trained on one task is reused and adapted for a related task, reducing the need for retraining from scratch. This saves time and data, especially in deep learning. [63].

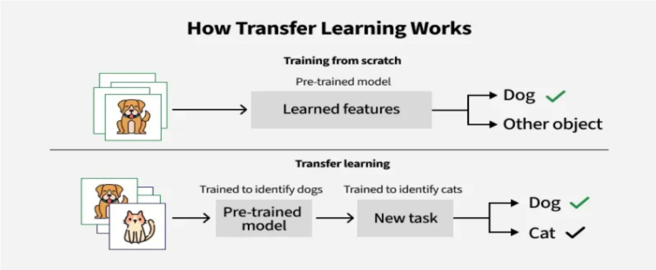


FIGURE 1.11: The architecture of Transfer Learning

- **Attention Mechanisms:** Is a deep learning technique that allows neural networks to focus on the most relevant information by weighting input elements differently. Inspired by the human brain, it improves the efficiency of models and is used in fields such as machine translation, computer vision, and natural language processing [64].

- **Autoencoders for unsupervised learning and feature extraction:**re neural networks used in unsupervised learning and feature extraction. They compress input data into a lower-dimensional representation via an encoder and then reconstruct it through a decoder, learning meaningful features by minimizing reconstruction error[65].

Advanced Techniques in Deep Learning

- **Classification:** Assigns labels to input data by learning patterns from features. CNNs are widely used for accurate image classification [66].

- **Object Detection:** Detects and locates objects in images or videos using bounding boxes. It merges classification and localization, with deep learning boosting its effectiveness in fields like surveillance and autonomous systems [67].

- **Segmentation:** Divides images into meaningful regions by labeling each pixel. CNN-based segmentation is essential in medical imaging, self-driving vehicles, and remote sensing [68].

FIGURE 1.12: Difference between classification,detection,segmentation [69]

1.7 Datasets

Data is essential for machine learning, as model performance relies on dataset quality. A diverse and accurate dataset enhances learning, leading to more reliable results.

1.7.1 Detecting floating objects above the dam water

The Marine Dataset from Roboflow, containing 6,424 images across six categories (ship, sinker, swimmer, trash, boat, buoy), was used to train the AI model. Its environmental diversity simulates real dam conditions, enabling accurate detection of floating objects and improving early monitoring and prevention s[70].

1.7.2 Crack detection

A combined dataset of 28,574 images was created by merging SDNET2018 with the dawgsurfacecrackssag dataset. It includes images of concrete structures, classified into two categories: with cracks and without cracks. This dataset was used to train the model for dam crack detection [71, 72].

1.7.3 Drowning detection

The Drowning Detection dataset from Roboflow, comprising 9,651 diverse images of people in water, was used to train a model for early drowning detection. It enables the system to analyze body positions and movements to predict potential drowning and trigger timely alerts [73].

1.8 Conclusion

This chapter provides a comprehensive overview of a smart dam monitoring system based on Internet of Things (IoT) and artificial intelligence (AI) technologies, highlighting the challenges associated with traditional systems and how to address them through modern technological solutions. The shortcomings of manual monitoring were addressed, particularly the difficulty of early detection of cracks or floating objects, and how AI can enhance the effectiveness of such monitoring through real-time intelligent monitoring. We also discussed the role of deep learning in analyzing images and videos to detect floating objects, structural cracks, and drowning, using advanced models such as YOLO and CNN. We also reviewed the types of sensors used to collect data, methods for transmitting it using technologies such as LoRaWAN and Zigbee, and processing mechanisms, both via cloud and edge computing. This chapter sets the stage for the practical aspects of the following chapters, which will review the smart models used, the results of the tests, and the extent to which this system can transform dam management, ensuring greater safety and sustainability in the face of environmental and structural challenges.

Chapter 2

Computer Vision , Deep Learning for Smart Detection

2.1 Introduction

With the rise of artificial intelligence and computer vision, it's now possible to detect and track objects automatically and accurately. This technology is being used in many areas, from security systems to self-driving cars and environmental monitoring. In the case of dams, floating objects like trash or small boats can cause serious problems for both operations and the environment. That's why using smart systems for object detection is becoming more important. This chapter introduces the technical background of our project, including basic concepts of computer vision, object detection, deep learning (especially YOLO), as well as data preparation, augmentation, and the tools we used during development.

2.2 Computer Vision

2.2.1 Definition of computer vision

Computer Vision, also known as Machine Vision, is a field that enables machines to interpret and understand visual information from images or videos. It involves object detection, localization, and tracking, aiming to simulate the human visual system. This technology is widely applied in areas like security, quality inspection, and autonomous driving[74].

2.2.2 The Importance of Computer Vision

Deep learning-based computer vision enables fast and accurate crack detection on dam surfaces, minimizing manual inspections and reducing potential risks. [75]. These systems allow continuous automated monitoring of water bodies, detecting waste that could obstruct flow or harm infrastructure. Advanced algorithms like YOLO and CNNs can analyze high-resolution images and detect even small defects, enabling proactive prevention.[76, 77].

2.3 Deep learning-based object detection algorithms

Object detection algorithms play a key role in analyzing visual data, with continuous advancements enhancing their accuracy and efficiency. This section highlights the main algorithms used for detecting floating objects and cracks in dams

2.3.1 Faster R-CNN (Region Convolutional Neural Network)

Faster R-CNNis an advanced object detection model that uses a Region Proposal Network (RPN) to generate candidate object regions. While it offers high accuracy, it is slower than YOLOv3 due to complex operations like region proposal and classification.

It also demands significant computational resources, making it less suitable for real-time or high-speed applications that require quick response times. [78].

2.3.2 RetinaNet

RetinaNet is a popular object detection model known for addressing class imbalance using Focal Loss, making it effective for detecting small or underrepresented objects. It balances speed and accuracy but may be less suitable for detecting larger floating objects like boats, where other models perform better[79].

2.3.3 YOLO (You Only Look Once)

YOLO (You Only Look Once)is a high-speed, real-time object detection model based on CNNs. Introduced in 2016 by Joseph Redmon et al. [80], it predicts bounding boxes and classes simultaneously through end-to-end training, achieving both fast inference and high accuracy [80]. Below is an image showing the architectural structure of the YOLO model

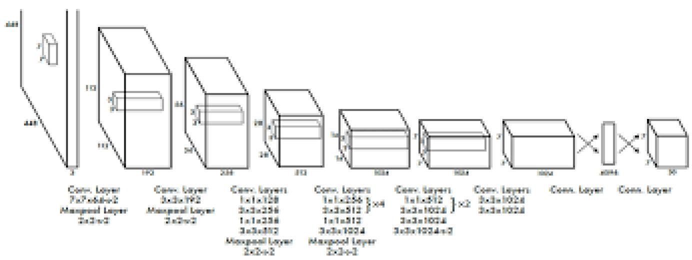


Figure 2.1: YOLO model architecture [81]

The model is divided into three main stages:

- **Image Division into Cells:** The first step in the YOLO algorithm is dividing the input image into a grid of cells. Each cell in this grid is responsible for detecting objects in its corresponding region. The size of the grid is denoted by $S \times S$ (e.g., 3x3 grid), and each cell is tasked with predicting the location and the class of the objects within its region[82].

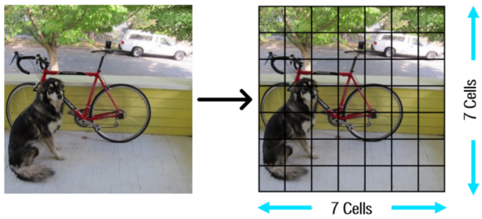


FIGURE 2.2: Divide the image into $(S \times S)$ grid

- **Bounding Box Prediction for Each Cell:** After dividing the image into cells, each cell predicts a set of B bounding boxes, where each box includes the following five components: x, y: the coordinates of the center of the bounding box relative to the grid cell. w, h: the width and height of the bounding box relative to the entire image. Confidence: a score that indicates the intersection over union (IoU) between the predicted bounding box and the ground truth box. Each grid cell, in essence, outputs a vector of predictions for each bounding box, and the confidence score helps evaluate how well the predicted box overlaps with the actual object in the image [80].

- **Intersection over Union (IoU):**Is a key metric in object detection used to evaluate the accuracy of predicted bounding boxes. It is the ratio between the overlapping area of the predicted and ground truth boxes to their combined area. A higher IoU means a better prediction. Typically, a threshold (e.g., 0.5) determines if a prediction is correct. IoU is crucial during both training and evaluation, as it helps assess bounding box confidence by comparing overlap with ground truth. [80].

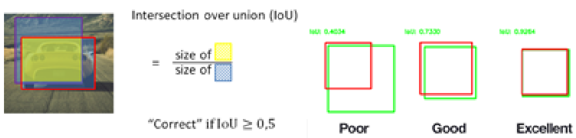


FIGURE 2.3: Intersection over union and example of IoU

- **Anchor Box:** After dividing the image into grid cells, each cell predicts a set of B bounding boxes. Each box includes five key components: x and y represent the coordinates of the center relative to the grid cell, w and h indicate the width and height relative to the entire image, and the confidence score reflects the Intersection over Union (IoU) between the predicted and actual ground truth box. These predictions form a vector that helps evaluate the accuracy of object localization within each cell.

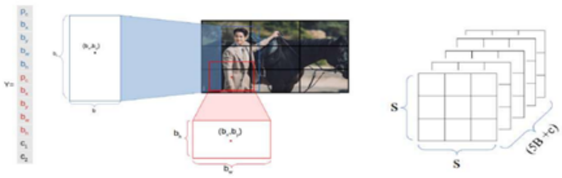


FIGURE 2.4: The predicted vector in the case of multiple boxes in the cell and the tensor that specifies the bounding box

- **Non Maximum Suppression:** In object detection, Non-Maximum Suppression (NMS) is used to eliminate duplicate bounding boxes for the same object by keeping only the one with the highest confidence score and removing the rest.[83]:
  - Start by selecting the bounding box with the highest confidence score.
  - Measure the Intersection over Union (IoU) between this box and the others.
  - Remove any box that has an IoU greater than 50% with the selected box.
  - Choose the next highest scoring box from the remaining candidates.
  - Continue this process until all bounding boxes have been reviewed, ensuring that only the most reliable detections are kept.

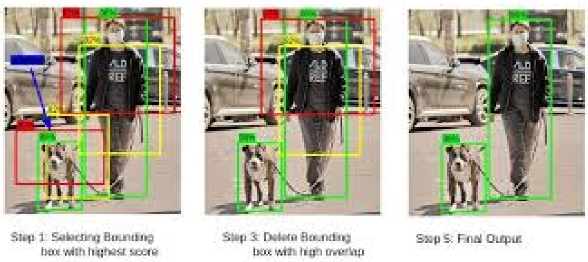


FIGURE 2.5: The output after different steps of NMS [84]

YOLOv2 Model

YOLOv2, developed in 2016 by Redmon and Farhadi, introduced key improvements such as batch normalization, anchor boxes, high-resolution classifiers, and multi-scale training. These enhancements boosted both speed and accuracy, achieving 76.8% mAP at 67 FPS on PASCAL VOC 2007, and 78.6% mAP with 544×544 resolution [85]. It also introduced YOLO9000, a unified model jointly trained on ImageNet and COCO, capable of detecting over 9,000 object categories, achieving 19.7% mAP on classes with detection labels and 16.0% on others, showcasing strong scalability [86].

YOLOv3 Model

YOLOv3, introduced in 2018, featured Darknet-53 as a new backbone with residual connections and a deeper architecture, improving feature extraction. It supported multi-scale predictions, boosting performance on objects of varying sizes. YOLOv3 achieved 57.9% mAP at IoU 0.5 with an inference time of 51 ms/image on a Titan X GPU, offering performance comparable to RetinaNet but with significantly faster speed—making it one of the most efficient detectors of its time [87].

YOLOv4 Model

YOLOv4, developed in 2020 by Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, achieved a balance between accuracy and speed in real-time object detection. It introduced key innovations such as Mish activation, CmBN, WRC, CSP connections, and Self-Adversarial Training (SAT). These enhancements enabled the model to reach 43.5% AP and 65.7% AP50 on MS COCO, running at 65 FPS on a Tesla V100 GPU. YOLOv4 was optimized for strong performance even on standard GPUs, making it ideal for real-time applications[88].

YOLOv5 Model

YOLOv5, released by Glenn Jocher (Ultralytics) in 2020 and implemented in PyTorch, improved upon YOLOv4 with a refined CSPDarknet53 backbone, enhanced neck components (SPPF, CSP-PAN), and a YOLOv3-style head. It introduced AutoAnchor to optimize anchor boxes per dataset and used advanced augmentation techniques like MixUp, Mosaic, and random affine transforms. Available in multiple sizes (YOLOv5n–YOLOv5x), it balances speed and accuracy, is open-source, and actively maintained by the community[89].

YOLOv6 Model

YOLOv6, released in 2022, introduced a redesigned architecture with components like the BiC module, SimCSPSPPF, and RepBi-PAN to boost detection efficiency. It combines anchor-based and anchor-free methods through Anchor-Aided Training (AAT) and improves small object detection via self-distillation. Available in versions (n, s, m, l), it achieves up to 52.8% AP. Like previous versions, it's open-source, actively maintained, and widely used in practice and research[90].

YOLOv7 Model

YOLOv7, developed in 2022 by Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, introduced key innovations like E-ELAN, planned re-parameterized convolution, coarse-to-fine label assignment, trainable bag-of-freebies. Trained from scratch on MS COCO (no pre-trained weights), it achieved 56.8% AP at 30+ FPS and 55.9% AP at 56 FPS (YOLOv7-E6) on an NVIDIA V100, outperforming models like SWIN-L and ConvNeXt-XL. It set a new standard in real-time object detection by balancing exceptional accuracy and high speed.[91].

YOLOv8 Model

YOLOv8, released by Ultralytics in January 2023, is an advanced real-time object detection model that builds on the YOLOv5 structure (backbone, neck, head) with major enhancements. It features an anchor-free detection head for improved accuracy and reduced errors, alongside optimized convolutional layers for better feature extraction.

The model supports tasks like instance segmentation and leverages FPNs for multi-scale object detection. Based on an enhanced Darknet-53, YOLOv8 achieves faster and more efficient performance than YOLOv7. It offers an easy-to-use API, enabling deployment on both embedded and cloud systems.

Available in variants like YOLOv8n, s, m, l, x, it provides flexibility across performance and resource needs. These improvements made YOLOv8 the model of choice in this project. [92]

2.4 Related works

2.4.1 Detecting Floating Objects and Drowning People

Ref	Year	Title	Approach	Dataset	Accuracy
[93]	2023	Soft-NMS YOLOv5 with SIOU for Small Water Surface Floater Detection	YOLOv5 enhanced with Soft-NMS and SIOU	Custom dataset	86.3% mAP

Ref	Year	Title	Approach	Dataset	Accuracy
[94]	2025	An annotated Dataset and Benchmark for Detecting Floating Debris in Inland Waters	YOLOv9, YOLOv8, YOLOv5s, Faster R-CNN, RetinaNet, SSD, CenterNet2	`IWHR_AI_Label_Floater_V1` (3000 images)	YOLOv9: 67.7%, YOLOv8s: 66.2%, YOLOv5s: 65.9%, Faster R-CNN: 54.6%
[95]	2023	Detecting Floating Objects on the Surface of Fresh Water Environments	YOLOv5-FF	FODS-00-01 (6240 images)	/
[96]	2024	MS-YOLO: A Lightweight and High-Precision YOLO Model for Drowning Detection	MS-YOLO	Indoor simulated drowning dataset	Precision: 80.9%, Recall: 7.8%, F1-Score: 82%, mAP@0.5: 86.4%
[97]	2023	Detection of Floating Objects on Water Surface Using YOLOv5s in an Edge Computing Environment	YOLOv5s	Special Dataset	mAP@0.5: 92.01%, mAP@0.5:0.95: 59.62%

Ref	Year	Title	Approach	Dataset	Accuracy
[98]	2024	Research on Detection of Floating Objects in River and Lake Based on AI Intelligent Image Recognition	SSD, Faster R-CNN, YOLOv5	Special Dataset	/
[99]	2021	Improved YOLO Based Detection Algorithm for Floating Debris in Waterways	FMA-YOLOv5s	Custom dataset (2400 river images)	mAP@0.5: 79.41%
[100]	2024	Improved YOLOv8 Algorithm for Water Surface Object Detection	YOLOv8n-MSS	Special Dataset	mAP@0.5: 87.9%, mAP@0.5:0.95: 47.6%
[101]	2024	Enhanced Water Surface Object Detection with Dynamic TaskAligned Sample Assignment and Attention Mechanisms	YOLOv8s	/	mAP: 47.5%

Ref	Year	Title	Approach	Dataset	Accuracy
[102]	2023	An Improved YOLOv5 Algorithm for Drowning Detection in the Indoor Swimming Pool	Enhanced YOLOv5 (ICA Module, BiFPN)	Drone-captured pool dataset	mAP@0.5: 98.5%, mAP@0.5:0.95: 73.3%
[103]	2024	Improved Automatic Drowning Detection Approach with YOLOv8	YOLOv8, YOLOv5 (for comparison)	Swimming and Drowning Detection Dataset	YOLOv8: mAP = 90.1%, YOLOv5: mAP = 88.5%
[104]	2025	WSEAS Transactions on Information Science and Applications	YOLOv8, ResNet50, Xception, CNN	Custom Dataset	YOLOv8: 82.1%, ResNet50: 83.4%, Xception: 85.8%, CNN: 66%
[105]	2023	An Improved YOLOv5 Algorithm for Drowning Detection in the Indoor Swimming Pool	Improved YOLOv5	Self-made dataset	Precision: 98.1%, Recall: 98.0%, mAP@0.5: 98.5%

Ref	Year	Title	Approach	Dataset	Accuracy
[106]	2024	Improved Automatic Drowning Detection Approach with YOLOv8	YOLOv8, YOLOv5	Swimming and Drowning Detection Dataset	YOLOv8: 90.1%, YOLOv5: 88.5%

Table 2.1: Recent studies on detecting floating objects and drowning cases.

The table above clearly demonstrates that YOLO models, in their various versions, are the preferred choice in modern research on floating object detection and drowning incidents. This is primarily due to their excellent balance of accuracy and real-time image processing capability. Several studies have found that newer versions like YOLOv8 and YOLOv9 perform better than older versions like YOLOv5, as well as traditional models like Faster R-CNN and SSD. Using custom datasets that reflect the true complexity of aquatic environments also helps these models perform better. For my project, which focuses on floating object detection in dams and early warnings of drownings, I chose YOLOv8 because it offers high accuracy, fast processing, and reliable results—key factors when working in sensitive environments where rapid responses are critical. YOLOv8 excels in the field of computer vision, outperforming older versions like YOLOv5 and slower models like Faster R-CNN. It achieves this through an anchorless detection head and enhanced convolutional layers that help the model better capture important features. Additionally, it uses feature pyramid networks (FPNs), which allow it to detect objects of various sizes, even in complex water scenes.

2.4.2 Detecting cracks in dams

Ref	Year	Title	Approach	Dataset	Accuracy
[107]	2024	Concrete Surface Crack Detection Algorithm Based on Improved YOLOv8	YOLOv8-CD (Improvements over YOLOv8n)	RDD2022 and Wall Crack	RDD2022: mAP50: 93.3%, mAP50-95: 77.6% Wall Crack: mAP50: 91.5%, mAP50-95: 77.6%
[108]	2020	Real-Time Concrete Crack Detection and Instance Segmentation using Deep Transfer Learning	YOLACT	500 manually annotated images from 5 open databases	mAP: 29.8%
[109]	2025	Data-driven Detection and Evaluation of Damages in Concrete Structures	YOLOv7, Mask R-CNN	/	YOLOv7: mAP@0.5: 96.1%, Speed: 40 fps Mask R-CNN: mAP@0.5: 92.1%, Speed: 18 fps

Ref	Year	Title	Approach	Dataset	Accuracy
[110]	2024	CL-YOLOv8: Crack Detection Algorithm for Fair-Faced Walls Based on Deep Learning	CL-YOLOv8 (with ConvNeXt V2 and LSKA)	Extended dataset of exposed wall surfaces	mAP: 83.7%
[111]	2025	Intelligent Road Crack Detection and Analysis Based on Improved YOLOv8	YOLOv8 + ECA + CBAM attention mechanisms	4,029 road surface images	/
[112]	2023	A Novel Approach for Concrete Crack and Spall Detection Based on Improved YOLOv8	YOLOv8	/	Accuracy: 94%
[113]	2024	Embedded Road Crack Detection Algorithm Based on Improved YOLOv8	YOLOv8 + C2f-Faster + SE Attention	RDD20 (Road Damage Detection 2020)	F1 Score: 57.3%

Ref	Year	Title	Approach	Dataset	Accuracy
[114]	2020	Detection Of Concrete Cracks using Dual-channel Deep Convolutional Network	CNN	Special data	Precision: 92.25%
[115]	2024	Bridge Bottom Crack Detection and Modeling Based on Faster R-CNN and BIM	Faster R-CNN + VGG-16	Drone-captured high-resolution bridge images	Precision: 92.03%, Recall: 96.54%
[116]	2023	Advanced Crack Detection and Segmentation on Bridge Decks Using Deep Learning	Faster R-CNN-ResNet50/101, RetinaNet-ResNet50/101, YOLOv7	/	mAP@0.5: 99.1%, mAP@0.5:0.95: 79.8% Precision: 98.7%, Recall: 95.1%, F1 Score: 0.97

Table 2.2: Recent studies on concrete crack detection cases.

Using the YOLO (You Only Look Once) model for crack detection is an excellent choice for several technical and practical reasons. First, YOLO boasts high image processing speed, detecting objects in an image at once without the need for multiple stages. This enables real-time or short-time inspection, essential for engineering applications that require continuous and efficient monitoring. Second, YOLO has good accuracy in locating small, scattered cracks on surfaces. The algorithm divides the image into a grid and accurately analyzes each part, allowing for detailed defect detection. Third, YOLO is easier to use and train than some other, more complex models, making it easier for researchers and engineers to set up and customize the model for their specific crack dataset. Finally, YOLO offers an excellent balance between speed and detection quality, making it suitable for use on devices with limited computing power while maintaining high accuracy. Several recent studies have supported this trend, with improved YOLOv8 models achieving significant performance, reaching accuracy rates above 90% on widely used datasets. Other variants, such as YOLOv7, have demonstrated high accuracy combined with fast inference speeds, outperforming more resource-intensive models like Mask R-CNN. Additional enhancements incorporating attention mechanisms have further boosted the model's ability to intelligently detect cracks on various surfaces. These results confirm the ability of modern YOLO models to provide accurate and rapid crack detection compared to other methods, making them a practical and effective choice that meets the dual requirements of accuracy and speed, which explains their widespread adoption in modern engineering research and applications.

2.5 Conclusion

In this chapter, we learned how to use deep learning techniques to detect objects, flooding, and cracks in dams. We chose the YOLOv8 model to perform these tasks due to its high detection capabilities in real time and with good accuracy. The next chapter will discuss the structure and features of this model in detail. Overall, using YOLOv8 is an effective step toward developing an intelligent system capable of quickly detecting threats or faults, enhancing safety and operational efficiency in dam-related environments.

Chapter 3

Conception

3.1 Introduction

In this chapter, we propose an integrated approach for detecting cracks, drowning incidents, and floating objects in dam environments. We will explore the intricate architecture of the YOLOv8 model, as well as the procedures for data preparation and model training. This chapter outlines the system objectives, architecture, and design steps.

3.2 System Goals

The goal of this project is to build a deep learning-based system that enhances dam safety monitoring by automatically detecting critical threats. The system targets three main tasks: detecting floating objects, identifying potential drowning situations, and recognizing cracks in dam concrete. A digital camera, functioning as an optical sensor, is used to periodically capture images of the dam's water surface and concrete structures. These images are analyzed by a pre-trained YOLOv8 model to identify objects or anomalies. This approach combines deep learning with IoT principles, offering a real-time, automated solution that improves the efficiency and accuracy of hazard detection in dam environments.

3.3 Architecture of the System

The deployed system follows a real-time operational architecture designed to detect hazards in dam environments using AI and IoT technologies. It begins with smart IoT cameras installed in critical locations to continuously capture live image streams. These visual inputs are processed by a pre-trained deep learning model (YOLOv8), which identifies potential risks such as floating objects, drowning individuals, or structural cracks. The detection results are sent to a centralized dashboard interface, where alerts are visually displayed. Each alert includes the location of the hazard on an interactive map, a snapshot of the incident, the detection time, and relevant metadata. This real-time interface helps dam operators monitor and respond to critical situations promptly and effectively.

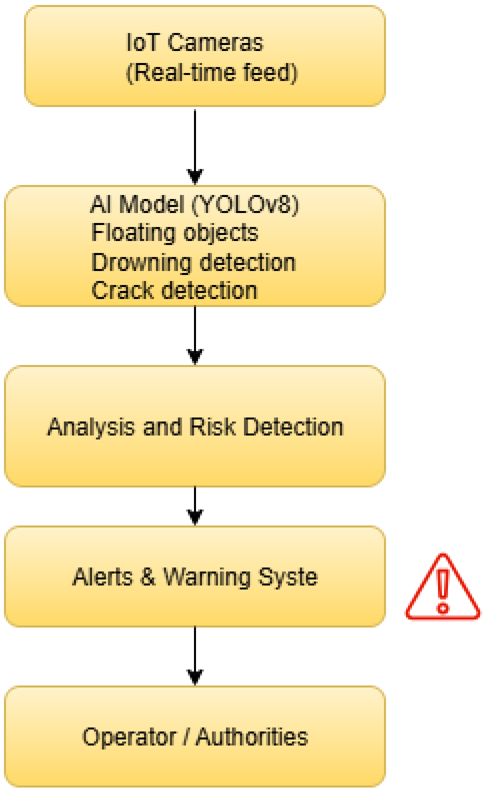


FIGURE 3.1: Global architecture

3.4 Data Acquisition

In our system, fixed digital cameras are strategically installed around the dam to periodically capture images of the water surface and concrete structures. This setup ensures stable, clear visuals that accurately reflect the environment. The captured images are directly analyzed by a pre-trained YOLOv8 model to detect floating objects, potential drowning cases, and cracks. This real-time visual data enables continuous monitoring and early hazard detection, enhancing overall dam safety..

3.5 Preprocessing and Model Training for Floating Object Detection

3.5.1 Data Collection

The dataset for floating object detection was carefully selected to ensure realism and diversity. The Marine Dataset from Roboflow was chosen for its richness, containing 14,278 images across six categories: boat (5,910), buoy (1,663), ship (3,356), sinker (535), swimmer (2,198), and trash (1,187). It is split into training (92.34%), validation (5.73%), and test (1.93%) sets. With accurate annotations and varied visual conditions (lighting, angles, environments), this dataset is ideal for robust model training in dam-like settings. [70].

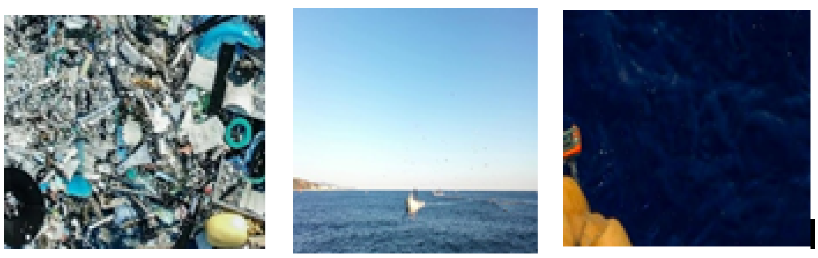


FIGURE 3.2: Some Pictures from Marine Dataset [70]

3.5.2 Data Preparation

Data Cleaning

A rigorous data cleaning process was applied to ensure quality and consistency. Low-quality, blurred, unlabeled, or incorrectly annotated images were removed, along with duplicates to avoid overfitting. Categories irrelevant to this task, such as ”sinker” and ”swimmer,” were excluded, as this part focuses on general floating object detection. These categories were addressed separately in Part 3 using a specialized drowning detection dataset.

Dataset Splitting

After cleaning, the dataset was restructured to ensure balanced learning. It was re-split into training (7,634 images – 79.99%), validation (955 images – 10.01%), and testing (955 images – 10.01%) subsets. This repartitioning guarantees fair sample distribution, enabling effective model training, reliable hyperparameter tuning, and proper evaluation on unseen data.

Challenges Faced in Data Preparation

During dataset analysis, we identified a class imbalance, with the trash category significantly underrepresented. To correct this without distorting class proportions, we applied a filtering step to select only images containing trash objects exclusively. This ensured that augmentation was applied in a controlled, class-specific manner. If we had augmented mixed-content images, it would have defeated the purpose of class balancing. After augmentation, trash instances increased from 929 to 1,730 objects across 1,465 images, improving the model's ability to detect this underrepresented class.

Augmentation

We applied the following transformations using the Albumentations library:

- **A.HorizontalFlip (p=0.5)**: Simulates orientation changes by flipping the image horizontally, reflecting real-world water movement.
- **A.RandomRotate90 (p=0.5)**: Introduces rotation to help the model recognize objects from different angles due to current or wind.
- **A.RandomBrightnessContrast (p=0.5)**: Adjusts lighting to reflect natural variations like sunlight, shadows, and reflections.
- **A.HueSaturationValue (p=0.5)**: Alters color properties to mimic differences in water quality, cameras, or time of day.
- **A.MotionBlur (p=0.3)**: Adds blur effects to simulate motion from waves or camera shifts, enhancing robustness.

It is important to highlight that augmentation was deliberately limited. The filtering process narrowed the selection to images containing only trash objects, which represented a small subset of the dataset. Expanding augmentation beyond this would have compromised class purity and potentially introduced unwanted noise or imbalance.

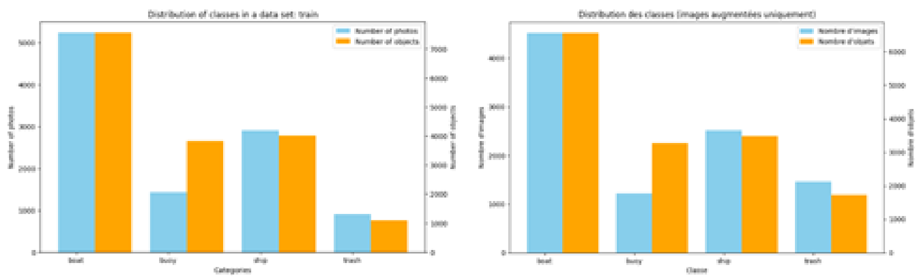


Figure 3.3: Distribution of categories before and after the augmentation process

3.5.3 Which Version Did We Choose and Why?

In this part, we chose the YOLOv8n model for the detection of floating objects in dam environments due to several technical and practical advantages:

- **Lightweight and Fast Inference**: YOLOv8n is the smallest and fastest model in the YOLOv8 family, making it ideal for real-time applications and deployment on edge devices with limited computational resources.
- **Strong Performance with Multi-Scale Objects**: The model incorporates FPN (Feature Pyramid Network) and PANet structures, enabling effective detection of objects at various scales—from very small to large—within the same image. This is especially important given that the dataset includes floating objects of different sizes.
- **High Accuracy Despite Simplicity**: YOLOv8n offers competitive accuracy despite its lightweight nature, thanks to architectural improvements such as C2f modules and better internal information flow.
- **Easy to Train and Deploy**: YOLOv8n integrates seamlessly with the Ultralytics framework, providing streamlined tools for training, validation, evaluation, and model export in formats such as ONNX or TorchScript.
- **Flexible and Customizable**: The model is highly adaptable to the specific needs of the project, including custom classes, image resolutions, and preprocessing strategies, allowing fine-tuned performance.
- **Strong Community Support and Reliability**: YOLOv8 benefits from an active and growing community, with extensive documentation, frequent updates, and community-driven troubleshooting, ensuring reliable and sustainable use.

3.5.4 Train Model on Our Data

Hyperparameter Choices to Train YOLOv8

The training process included two main phases: a warm-up stage focused on the detection head, followed by fine-tuning of the entire network. In both phases, key hyperparameters were carefully selected to match the dataset characteristics and the challenges of detecting floating objects. The table below summarizes these hyperparameters, their values, and selection rationale.

Hyperparameter	**Value Phase 1**	**Value Phase 2**	**Explanation & Justification**
epochs	90	180	Balanced between learning progression and overfitting prevention.
patience	20	20	Early stopping patience (if no improvement).
imgsz	640	768	Larger image size helps better detect small floating objects.
batch	16	32	Batch size adapted to memory, larger in Phase 2.
optimizer	Adam	Adam	Adaptive optimizer.
lr0	1e-4	5e-5	Initial learning rate.
lrf	0.01	0.001	Final learning rate ratio.
weight_decay	0.001	0.001	Regularization parameter.
cos_lr	True	True	Cosine LR schedule.
freeze	[0–9]	[]	Frozen layers in phase 1.
mosaic	0.5	0.5	Mosaic augmentation.
mixup	0.1	0.1	Mixup augmentation.
hsv_h	0.015	0.015	Hue shift.
hsv_s	0.7	0.7	Saturation shift.
hsv_v	0.4	0.4	Brightness variation.

Hyperparameter	Value Phase 1	Value Phase 2	Explanation & Justification
translate	0.1	0.1	Translation augmentation.
scale	0.3	0.3	Scale augmentation.
degrees	10.0	10.0	Rotation augmentation.
fliplr	0.5	0.5	Horizontal flip.
save	True	True	Save checkpoints.
save_period	10	10	Save every 10 epochs.
exist_ok	True	True	Overwrite existing folders.

TABLE 3.1: YOLOv8 Training Configuration for Detection of Floating Objects

3.6 Preprocessing and Model Training for Detection Cracks in Dams

3.6.1 Data Collection

Selecting a suitable dataset for crack detection in dams was challenging, as it required the inclusion of both crack and crack-free (void) images to ensure balanced training and reduce false positives. To meet this need, we combined two complementary datasets. The SDNET2018 dataset from Kaggle, comprising 56,092 images, was used as the main source for crack-free samples. From its three surface classes, we selected the one that best resembles dam concrete textures to ensure realism. For cracked images, we used dawgsurfacecrackssag, available on Roboflow, which includes 23,008 high-quality, labeled crack images. The merging of both datasets resulted in a balanced dataset of 28,574 images, equally split between crack and no-crack categories (14,287 each) [71, 72]

FIGURE 3.4: Some Pictures from dawgsurfacecrackssag Data [71, 72]

3.6.2 Data Preparation

Dataset Merging

To create a balanced dataset for crack detection, we merged two sources: dawgsurfacecrackssag (for cracked images) and SDNET2018 (for non-cracked surfaces). Both datasets were standardized in format and labeling. This combination allowed the model to better distinguish between cracked and intact areas, enhancing detection accuracy and reducing false positives.

Balancing the Dataset by Synchronizing the Number of Images per Class

To address class imbalance, the dataset was balanced by reducing the size of the larger class to match the smaller one. This step ensured equal representation of cracked and non-cracked images during training, improving the model's generalization and reducing the risk of overfitting to the majority class.

Divide Data

After merging the two datasets, the unified set was split into training (70.01%), validation (19.98%), and testing (10.01%) subsets, totaling 20,001, 5,714, and 2,859 images, respectively. Annotation files were carefully aligned with their corresponding images to preserve data integrity and ensure accurate model training and evaluation.

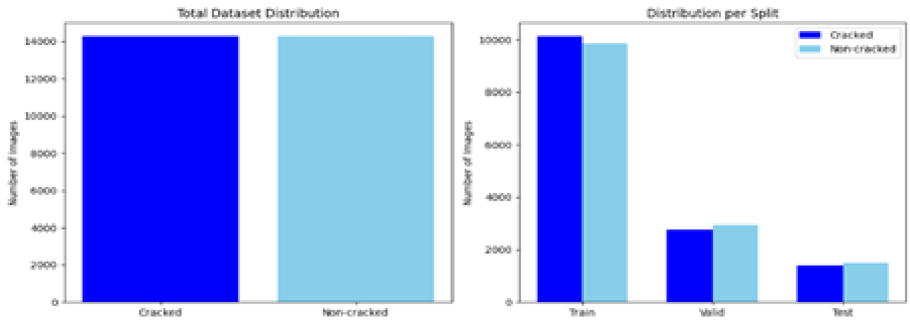


FIGURE 3.5: Distribute Images for the Combined Dataset (Detection Cracks)

Challenges Faced in Data Preparation

- **Absence of Tag Files in Background Images**: Images without cracks were not accompanied by tag files, while detection models require a tag file for each image. Blank files were created for these images to maintain data structure consistency.
- **Maintaining Image and Label Consistency During Segmentation**: When segmenting the data into training, validation, and test sets, it was essential to ensure that each image was transmitted correctly with its label file to avoid any loss of information.

3.6.3 Which Version Did We Choose and Why?

In this study, the YOLOv8-Nano model was selected for crack detection tasks due to several key advantages:

- **Lightweight and Real-Time Capabilities**: YOLOv8-Nano is a highly efficient, lightweight model (< 5MB), designed for fast inference on low-power or embedded devices.
- **Superior Detection of Small Cracks**: Cracks are often thin, faint, and difficult to detect. YOLOv8 introduces anchor-free heads, refined receptive fields, and enhanced multi-scale feature extraction.

- **Scalability to Large and Irregular Cracks**: Thanks to its improved neck architecture and feature fusion (FPN/PAN), YOLOv8n can accurately detect long, irregular surface cracks.
- **Fast Training and Easy Customization**: Due to its small size and optimized architecture, YOLOv8n trains quickly and adapts well to new datasets.
- **Modern Features and Robust Performance**: YOLOv8n includes modern enhancements such as improved frame resolution handling.
- **Edge Deployment Ready**: Its minimal size and computational efficiency allow YOLOv8n to be deployed on low-resource devices without needing cloud infrastructure.
- **Performance Comparison with Larger Models**: Although a larger model such as YOLOv8-M was also tested, it yielded slightly lower performance in this specific task, with a mAP50 of 78% and mAP50-95 of 92%, compared to the higher performance achieved by YOLOv8-Nano.

3.6.4 Train Model on Our Data

Hyperparameter Choices to Train YOLOv8-n

Hyperparameter	Value	Explanation & Justification
epochs	50	Provides sufficient training while avoiding overfitting.
patience	10	Allows early training to be stopped if the model is not improving.
imgsz	640	Image size that balances resolution and memory consumption.
batch	16	Batch size compatible with available memory on GPU.
optimizer	Adam	To achieve stable convergence speed using adaptive learning rate.
lr0	0.001	A moderate initial learning rate speeds up model learning.
lrf	0.01	A low end learning rate ratio helps to gradually reduce the learning rate.
warmup_bias_lr	0.1	A custom learning rate for biases at the beginning of training.

Table 3.2: YOLOv8 Training Configuration for Detection of Cracks in Dams

3.7 Preprocessing and Model Training for Detection Drowning at Dams

3.7.1 Data Collection

For the drowning detection task, the publicly available "Drowning Detection" dataset from Roboflow was used. Originally composed of three categories (non-swimming, swimming, drowning), the dataset was refined to include only swimming and drowning categories, excluding non-swimming due to its limited diversity and low relevance. After filtering, 9,046 images remained and were split as follows: 70% for training (7,197 images), 15% for validation (1,099 images), and 15% for testing (750 images)[73].[73].

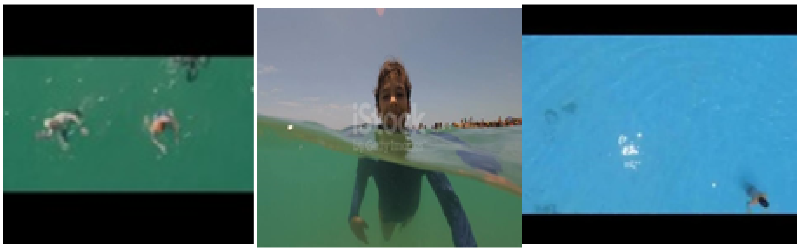


FIGURE 3.6: Some Pictures from Drowning Detection Dataset [73]

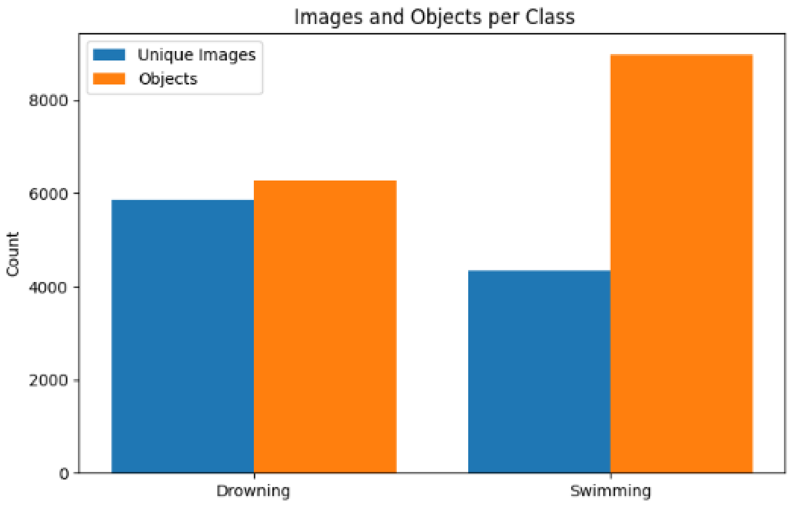


FIGURE 3.7: Distribute Images and Objects into Categories

3.7.2 Data Preparation

Data Cleaning

As part of preprocessing, a data cleaning step was performed to ensure the quality and relevance of the drowning detection dataset. Images lacking label files or containing incorrect annotations were removed. Only properly labeled images with Swimming and Drowning instances were kept. The validated data were then organized under a cleaned-dataset structure, maintaining the original train/val/test split.

Challenges Faced in Data Preparation

- **Class Imbalance and Irrelevant Labels**: The “not swimming” class was underrepresented and included irrelevant data. It was excluded to avoid noise.
- **Cleaning and Filtering the Labels**: Scripts were used to remove annotations related to the excluded class and adjust remaining labels.
- **Maintaining Dataset Consistency**: Ensuring dataset integrity after modifying label files was critical.
- **Class Re-indexing and YAML Configuration**: Class indices were updated consistently across all splits, and the data.yaml file was modified.
- **Verifying Final Distribution**: Confirming the final image and object distribution per class was essential.
- **Avoiding Data Leakage and Split Validation**: Care was taken to ensure proper data splitting ratios without overlap.

3.7.3 Which Version Did We Choose and Why?

For the task of detecting drowning in aquatic environments, the YOLOv8-M model was chosen for the following reasons:

- **Higher Accuracy**: YOLOv8-M outperformed both YOLOv8-N and YOLOv8-S models in training.
- **Better Generalization in Aquatic Environments**: The model effectively handles complex aquatic scenes.
- **Reliable Detection of Isolated Objects**: YOLOv8-M has a better ability to detect partially submerged or partially visible individuals.
- **Balanced Performance**: It provides an ideal balance between computational cost and detection accuracy.

3.7.4 Train Model on Our Data

Hyperparameter Choices to Train YOLOv8-m

Hyperparameter	Value	Explanation & Justification
epochs	50	Provides enough training cycles for effective learning without overfitting.
imgsz	640	A standard input size that balances detection quality and resource usage.
batch	16	Suitable for available GPU memory in Kaggle.
optimizer	SGD	Preferred for better generalization and stable learning.
lr0	0.001	An appropriate initial learning rate for a medium-sized model.
lrf	0.01	A suitable final learning rate factor that allows gradual convergence.
amp	True	Enables mixed-precision training to reduce memory usage.
patience	10	Enables early stopping if no improvement is seen over 10 epochs.
workers	2	Set low to avoid data loading issues in Kaggle.

Table 3.3: YOLOv8 Training Configuration for Detection of Drowning Incidents

3.8 Training Constraints and Challenges

During the training of our models, we encountered several constraints that significantly impacted the workflow and training efficiency:

- **Google Colab Limitations**: GPU sessions were unstable and frequently disconnected.
- **Session Expiration Before Completion**: Both in Colab and Kaggle, session time limits terminated training prematurely.

- **Transition to Kaggle**: Kaggle provided more consistent resources, but training still took considerable time.
- **Time Constraints**: Managing large datasets and training deep learning models requires significant computational time.
- **Limited Hyperparameter Experimentation**: GPU memory limitations restricted experimentation with certain hyperparameters.

3.9 Model Configuration

3.9.1 Why YOLO Was Chosen

- **High Inference Speed for Real-Time Applications**: YOLO offers high-speed inference, crucial for water surface monitoring.
- **Excellent Balance Between Accuracy and Speed**: YOLO offers a favorable trade-off compared to region-based detectors.
- **Lightweight and Edge-Device Friendly**: Optimized for deployment on low-resource devices.
- **Ease of Customization and Retraining**: Supports fast fine-tuning on custom datasets.
- **All-in-One Framework with Developer Support**: Provides a complete object detection pipeline with strong community support.
- **Improved Detection of Small and Overlapping Objects**: Enhances performance on small and partially occluded objects.
- **Active Community and Continuous Development**: Ensures regular updates and access to learning resources.

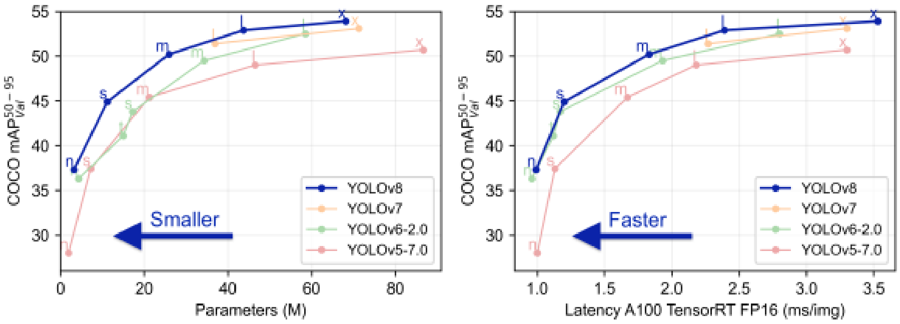


FIGURE 3.8: Comparing YOLOv8 to Other YOLO Models: A Comparative Analysis [117]

3.9.2 YOLOv8 Architecture

YOLOv8 is the latest evolution in the YOLO family of models, built to handle complex computer vision tasks such as object detection, image classification, and instance segmentation. The model benefits from a redesigned backbone, neck, and head, integrating more efficient convolutional layers. YOLOv8 adopts an anchor-free detection head, which simplifies bounding box prediction and enhances detection precision. It also supports instance segmentation, allowing differentiation between multiple instances of the same class within a single image. The backbone, inspired by the Darknet-53 architecture, has been optimized to extract more detailed features through larger feature maps. Additionally, it incorporates Feature Pyramid Networks (FPNs), enabling robust performance when detecting objects of diverse sizes in a single frame.

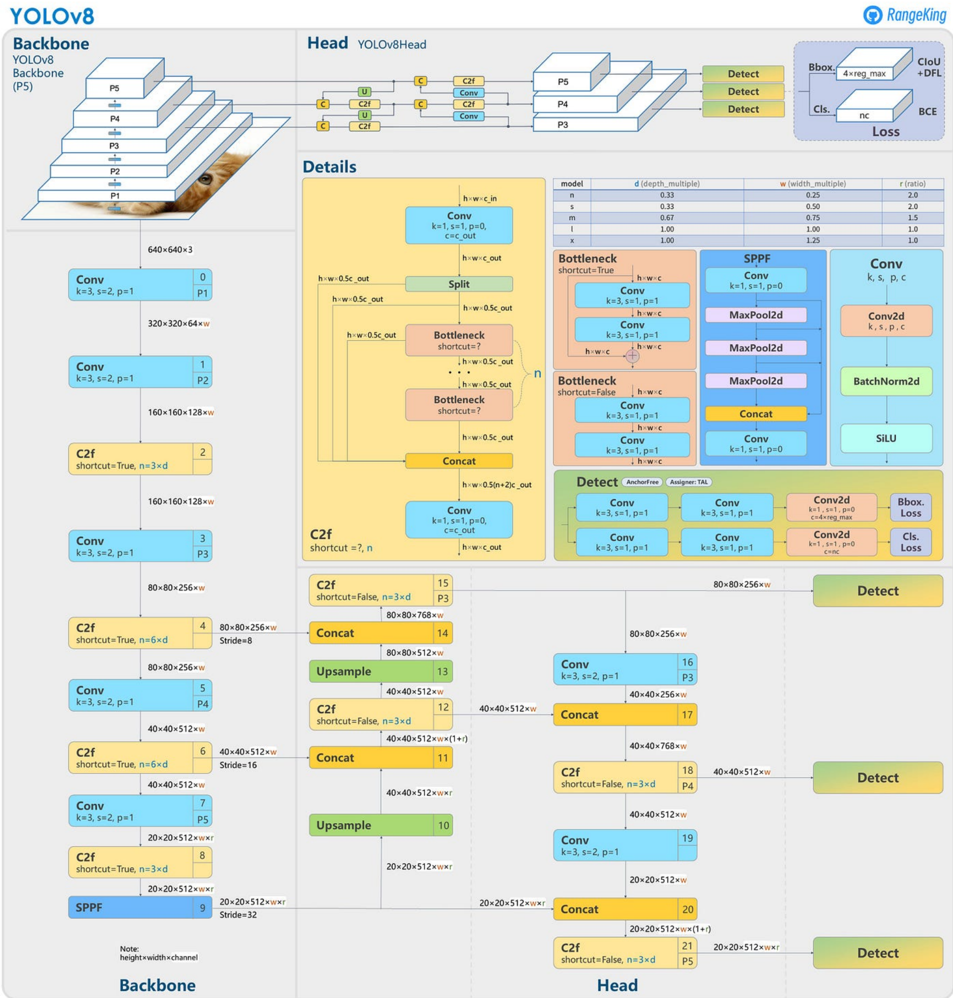


FIGURE 3.9: YOLOv8 Architecture [117]

3.10 Conclusion

In conclusion, this chapter provided a detailed examination of the training process for the YOLOv8 model within the context of our dam monitoring system. We explored the system's architecture, the data preparation for each of the three detection tasks—floating object detection, drowning detection, and crack detection—as well as the specific setup and training procedures of the model. This chapter lays the foundation for understanding the technical implementation, with the resulting performance and analysis to be presented in the next chapter.

Chapter 4

Implementation

4.1 Introduction

An integrated development environment (IDE) including software and hardware tools was adopted to design and implement the smart dam monitoring system. In this project, we used a set of programming languages, frameworks, and libraries appropriate for each task, along with an execution environment that supports efficient training and testing.

In this chapter, we will review the components of the development environment used, including hardware and software, then explain the steps for implementing the system and the techniques adopted at each stage. We will also present the most significant challenges we faced during development, and finally present the results obtained to evaluate the model's performance in each task.

4.2 Development Environment

4.2.1 Characteristics of the Material Used

Model Part	Used Laptop
Processor	Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz
RAM	8.00 GB
System type	64-bit Operating System, x64-based processor
Edition	Windows 10 Pro

Table 4.1: Characteristics of the material used.

4.2.2 Hardware Environment

Google Colaboratory: Google Colab is a free cloud-based platform that lets users write and run Python code directly in the browser with no setup required. It supports free GPU and TPU usage, making it ideal for machine learning and data science. As an extension of Jupyter Notebooks, it stores files on Google Drive and facilitates easy sharing and collaboration. [118]. **Kaggle**: Kaggle is Google's advanced cloud platform for data science and machine learning. It offers an interactive, browser-based environment for developing and testing models without local setup. Users can upload datasets, use Jupyter Notebooks, and access GPUs like NVIDIA Tesla P100/T4 and TPUs (e.g., TPU v3-8) to accelerate deep learning tasks. The platform provides 13 GB RAM, up to 70 GB cache, and 30 free GPU hours per week. With support for CPUs and a strong global community, Kaggle is ideal for experimentation, learning, and participating in data science competitions.

4.2.3 Software Environment

Python: Python is a versatile and easy-to-learn programming language known for its clear syntax, dynamic typing, and strong support for object-oriented programming. As an interpreted language, it enables rapid development and is widely used in both education and professional fields due to its readability, flexibility, and rich standard library[119]. **PyTorch**: PyTorch is an open-source deep learning framework written in

Python, known for its flexibility and ease of use. It supports GPU acceleration, reverse-mode auto-differentiation, and dynamic computation graphs, making it ideal for tasks like image recognition and language processing, as well as rapid experimentation and prototyping [120].

Jupyter Notebook: Jupyter Notebook is a free, open-source web-based tool for creating and sharing interactive computational documents. Originally called IPython Notebook, it was renamed in 2014. It supports over 40 programming languages, including Python, R, and Scala, making it ideal for both education and research. [121]. **OpenCV**: Open Source Computer Vision Libraryis an open-source library used for image processing, computer vision, and machine learning. It offers over 2,500 optimized algorithms for tasks like object recognition, face detection, 3D vision, robotics, and augmented reality. Supporting multiple languages (Python, C++, Java) and operating systems, it is a powerful and flexible tool widely adopted in AI and computer vision research [122]. **Matplotlib**: Matplotlib is a Python library for data visualization and graphical plotting, supporting various platforms. It enables the creation of different types of charts like histograms, scatter plots, and bar graphs. When used with NumPy, it becomes a powerful tool for scientific and numerical visualization. As an open-source alternative to MATLAB, Matplotlib offers flexible APIs that allow developers to integrate plots into GUI applications [123]. **Albumentations**: Albumentations is a high-performance image augmentation library designed to enhance deep learning models in computer vision. It offers a variety of transformations—like rotation, flipping, brightness changes, and noise addition—to boost training data diversity, especially for small or imbalanced datasets. Widely used in industry and research, it integrates smoothly with frameworks like PyTorch and TensorFlow, helping models generalize better across diverse data[124].

Streamlit: Is an open-source Python framework that allows data scientists and AI/ML engineers to rapidly build interactive data applications with minimal code. It enables quick development and deployment of dynamic apps, making it as simple to use as installing a standard Python library. [125]. **Folium**: Folium is a powerful Python library that combines Python's data processing with Leaflet.js's mapping capabilities.

It enables the creation of interactive maps and geospatial visualizations, which can be shared as HTML files or embedded in web apps—making it ideal for analyzing and displaying geospatial data.[126].

4.3 Training and Validation

The model has been trained and is now in the validation phase to evaluate its performance. To ensure the highest level of efficiency, the training process was continuously monitored. We will review the training results according to the project's three main axes: detecting floating objects, monitoring flooding, and detecting dam cracks.

4.3.1 Training and Validation Visualizations

In this section, we present selected examples of validation training images in each task:

4.3.2 Training and Validation Visualizations

1. **Detection of Floating Objects**

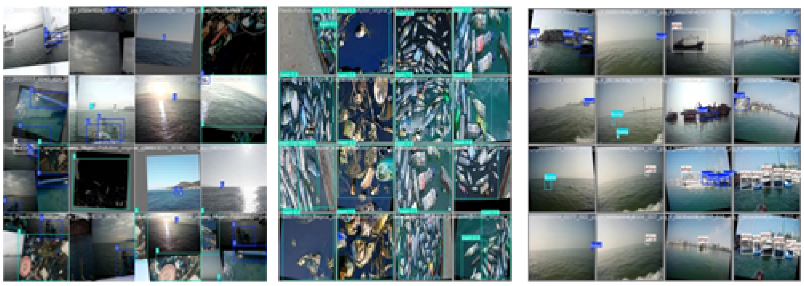


FIGURE 4.1: Train and Validation batches (floating objects).

2. **Detection of Cracks in Dams**

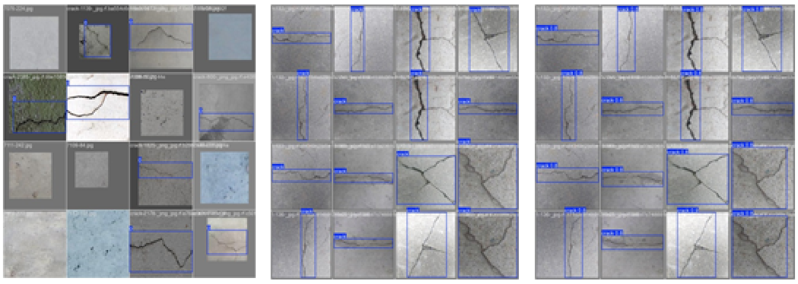


FIGURE 4.2: Train and Validation batches (cracks in dams).

3. **Detection of Drowning at Dams**

FIGURE 4.3: Train and Validation batches (drowning at dams).

4.3.3 Train/Val Metrics

Metrics used for evaluation include:

- *True Positives (TP)*: Instances where the model correctly identifies a positive sample.
- *False Positives (FP)*: Instances where the model incorrectly classifies a negative sample as positive.
- *True Negatives (TN)*: Instances where the model correctly identifies a negative sample.
- *False Negatives (FN)*: Instances where the model incorrectly classifies a positive sample as negative.

- *Mean Average Precision (mAP)*: Measures the model's capacity to reliably recognize objects.

- *Intersection over Union (IoU)*: Estimates the overlap between predicted and ground truth bounding boxes.

- *Precision*:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- *Recall*:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

1. **Detection of Floating Objects** The model was trained using a two-stage strategy: freezing the first 10 layers for 90 epochs, then fine-tuning all layers for 180 epochs.

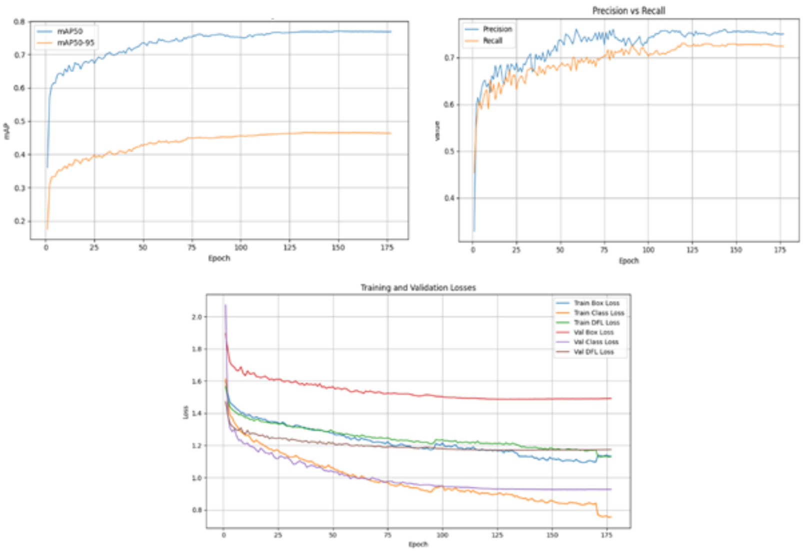


FIGURE 4.4: Results of floating objects.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	955	1828	0.756	0.728	0.77	0.466
boat	566	794	0.788	0.816	0.859	0.505
buoy	158	480	0.676	0.487	0.535	0.218
ship	315	447	0.727	0.776	0.797	0.43
trash	95	107	0.834	0.832	0.887	0.709

TABLE 4.2: Results of train set of floating objects.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	955	1828	0.75	0.727	0.761	0.471
boat	566	794	0.777	0.805	0.844	0.481
buoy	158	480	0.637	0.465	0.512	0.224
ship	315	447	0.703	0.727	0.75	0.393
trash	95	107	0.884	0.913	0.937	0.788

TABLE 4.3: Results of test set of floating objects.

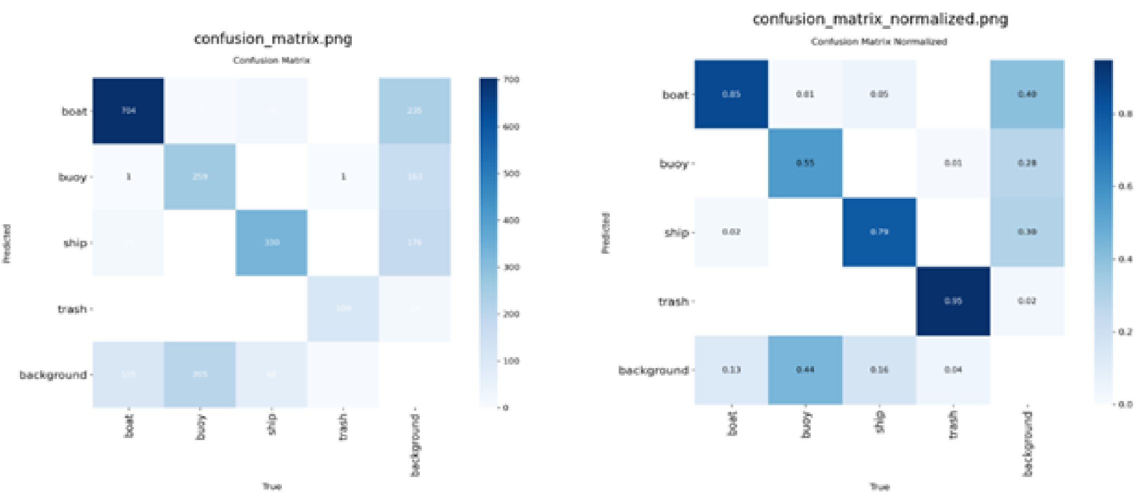


FIGURE 4.5: Confusion matrix and Matrix Normalized (floating objects).

2. **Detection of Cracks in Dams** The model was trained on 50 epochs.

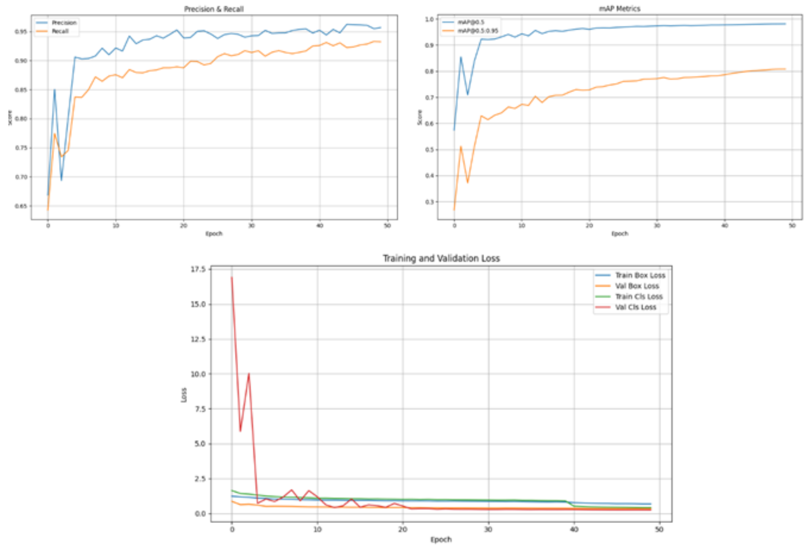


FIGURE 4.6: Results of cracks in dams.

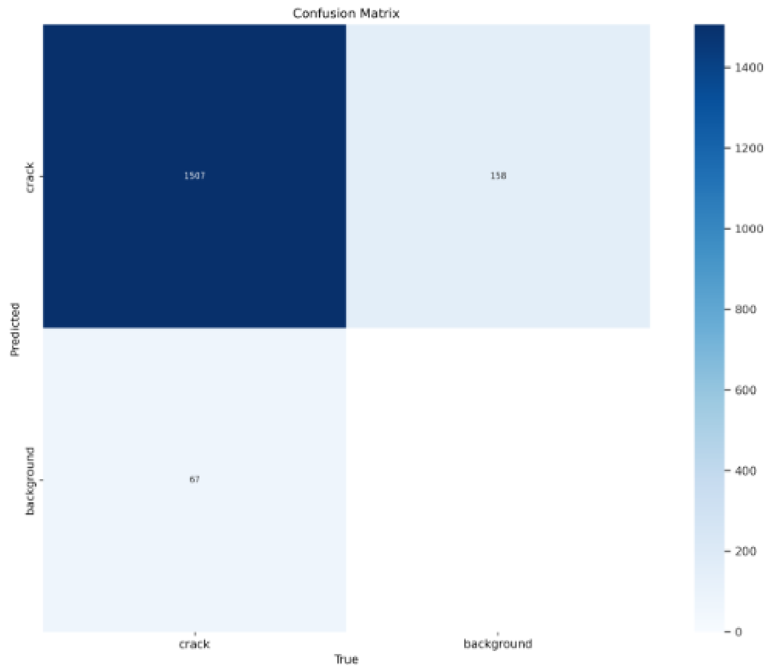


FIGURE 4.7: Confusion matrix for cracks in dams.

Dataset	Images	Instances	Precision	Recall	mAP50	mAP50-95
Train set	5714	3027	0.955	0.935	0.981	0.828
Test set	566	794	0.971	0.923	0.974	0.802

TABLE 4.4: Results of test and train set for cracks in dams.

3. **Detection of Drowning at Dams** The model was trained on 50 epochs.

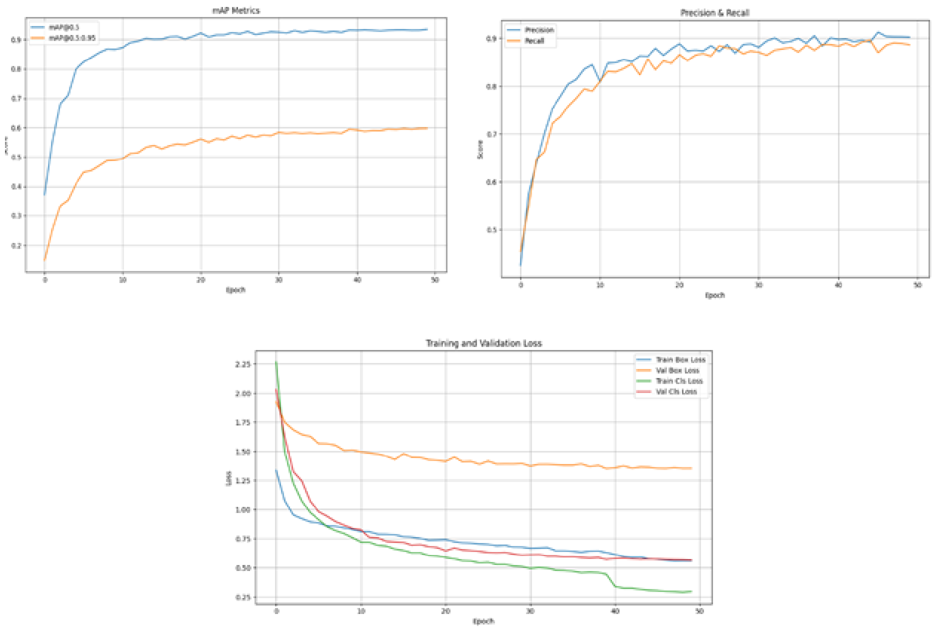


FIGURE 4.8: Results of drowning at dams.

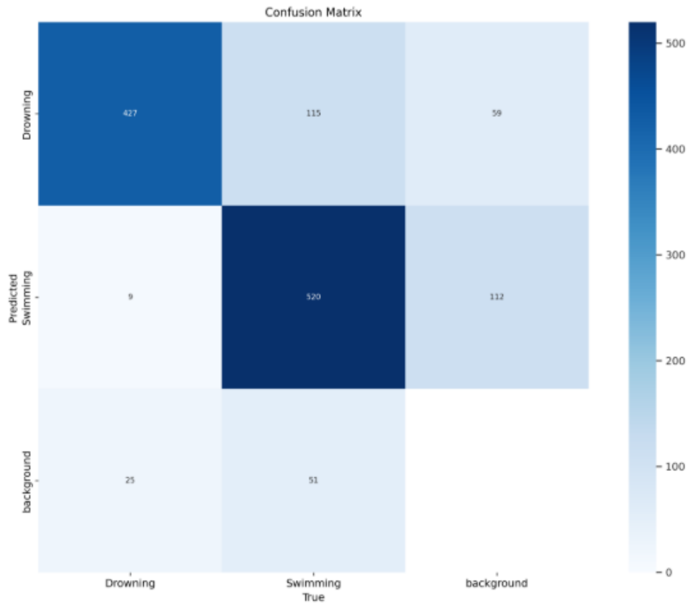


FIGURE 4.9: Confusion matrix of drowning at dams.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	1099	1959	0.902	0.887	0.935	0.598
Drowning	720	788	0.914	0.911	0.946	0.634
Swimming	553	1168	0.891	0.862	0.924	0.561

Table 4.5: Results of train set of drowning at dams.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	750	1147	0.807	0.801	0.809	0.517
Drowning	433	461	0.751	0.889	0.78	0.539
Swimming	394	686	0.864	0.712	0.838	0.494

Table 4.6: Results of test set of drowning at dams.

4.4 Test, Results, and Discussion

4.4.1 Test Results

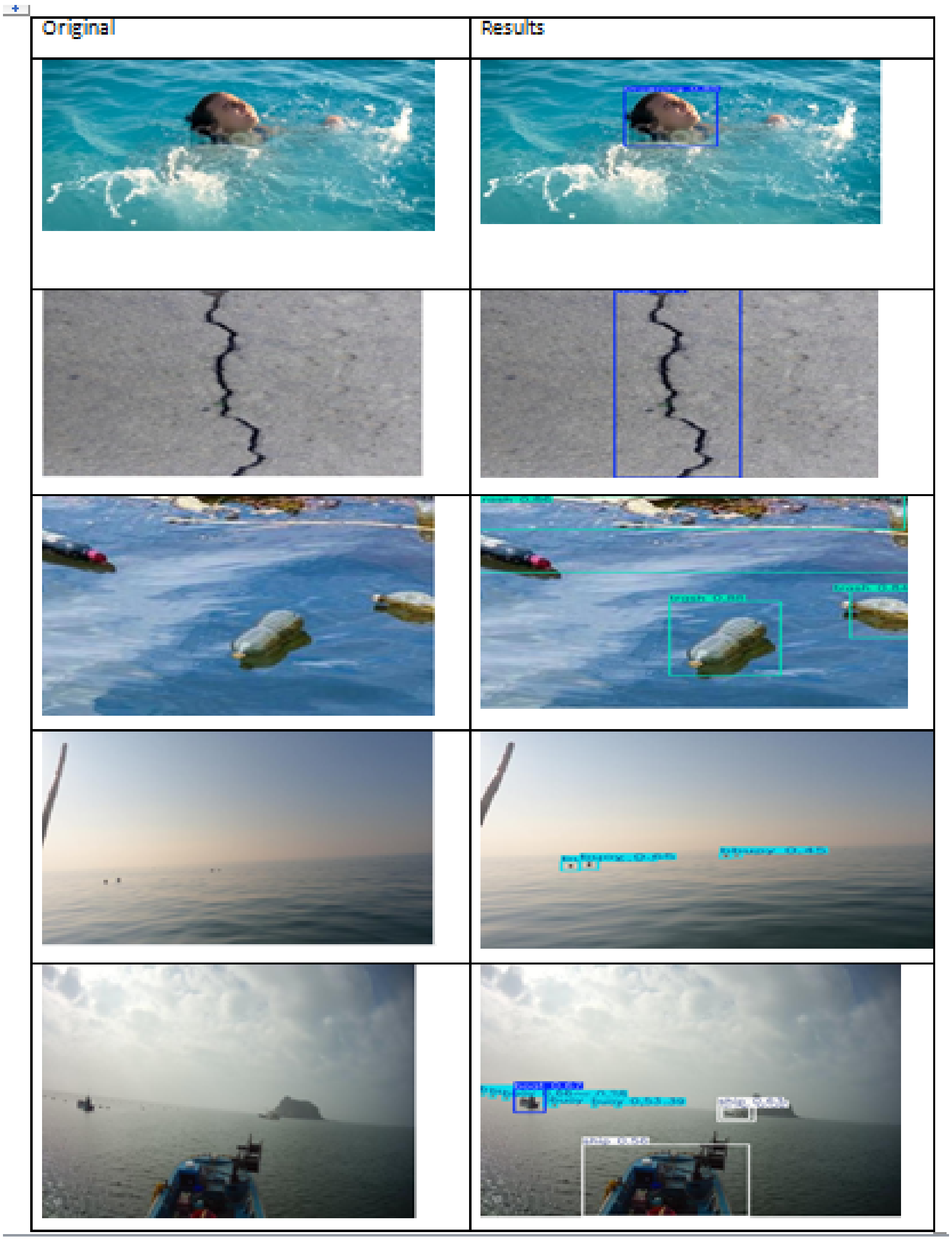


FIGURE 4.10: Inference Results on Test Images.

Tasks	Floating Objects		Drowning			Cracks
Model	Yolov8n	Yolov8m	Yolov8n	Yolov8s	Yolov8m	Yolov8n
mAP50	0.77	0.68	0.905	0.935	0.935	0.981
mAP50-95	0.466	0.38	0.54	0.579	0.598	0.828
Recall	0.728	0.63	0.85	0.886	0.887	0.935
Precision	0.756	0.67	0.867	0.907	0.902	0.955

Table 4.7: Training models results.

Tasks	Floating Objects		Drowning			Cracks
Model	Yolov8n	Yolov8m	Yolov8n	Yolov8s	Yolov8m	Yolov8n
mAP50	0.466	0.41	0.766	0.787	0.809	0.974
mAP50-95	0.77	0.72	0.467	0.488	0.517	0.802
Recall	0.727	0.637	0.775	0.815	0.801	0.923
Precision	0.75	0.668	0.763	0.789	0.807	0.971

Table 4.8: Test models results.

4.4.2 Discussion

The results obtained across the three core tasks—floating object detection, drowning detection, and crack detection—highlight the ability of the YOLOv8 architecture to adapt to diverse visual environments and operational constraints. Each task brought forth unique challenges, ranging from small-scale object variability to complex human movement patterns and fine-grained texture analysis.

In the *floating object detection* task, the use of YOLOv8n resulted in a more favorable balance between detection accuracy and consistency. It achieved a test mAP50 of 0.466 and a mAP50-95 of 0.77, with a precision of 0.75 and recall of 0.727. While the overall detection of objects such as trash and boat remained strong, categories like buoy—typically smaller and visually ambiguous—posed greater challenges. Comparatively, the YOLOv8m model showed lower generalization capacity, with a test mAP50 of 0.41, suggesting that the lightweight version (YOLOv8n) handled the task more effectively, particularly in scenarios involving small object sizes and cluttered water surfaces.

For *drowning detection*, performance was significantly enhanced with the use of the YOLOv8m model, which yielded a test mAP50 of 0.809, mAP50-95 of 0.517, precision of 0.807, and recall of 0.801. These results were further supported during training, where the model reached a recall of 0.887 and mAP50 of 0.935. Such metrics indicate high responsiveness in identifying early-stage drowning behaviors, despite the visual complexity of dynamic human movement in water.

The *crack detection* task consistently exhibited the highest performance across all evaluations. The YOLOv8n model achieved a training mAP50 of 0.981 and test mAP50 of 0.974, alongside exceptional precision (0.971) and recall (0.923) in the test phase. These values reflect the model's strong capacity to learn and generalize thin, high-contrast patterns across varied surfaces.

In summary, the results reflect how different configurations of YOLOv8 excel in different contexts: YOLOv8n for small-scale object detection in floating debris scenarios, YOLOv8m for capturing complex human-water interactions in drowning detection, and YOLOv8n again for high-precision structural crack analysis.

4.5 System Interface

The main interface allows users to choose the type of detection by displaying a drop-down list of the three types.

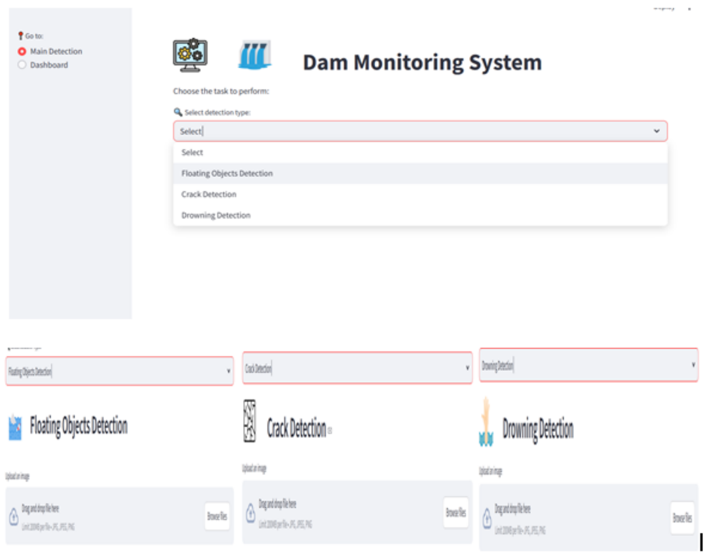


FIGURE 4.11: Interface main of our system.

The home page also contains a dashboard button that takes users to another page showing the hazard's location, type, time, day, and type on a map. It also shows the number of detected cases across the system's three tasks.

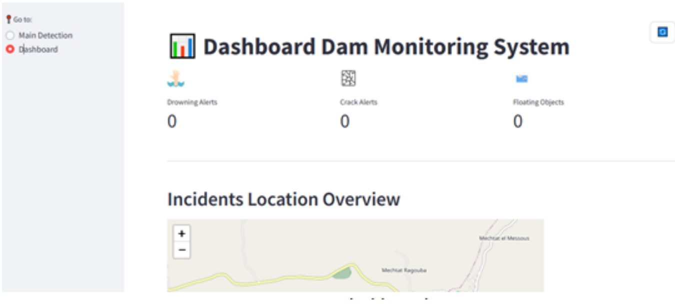


FIGURE 4.12: Dashboard page.

The latter contains a Reset button allows you to reset or restore the initial state of the interface. Below is a demonstration of how the system works to detect floating objects and react to alerts in practice Here we put a pucture with waste:

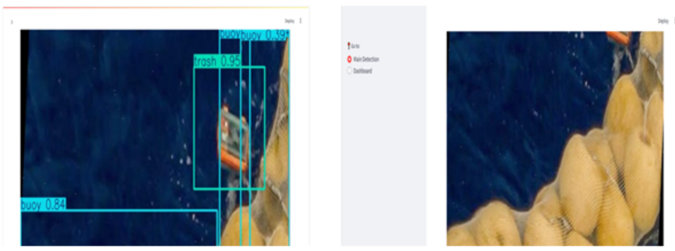


FIGURE 4.13: Detection Process (objects).

Here we see that it actually appeared in the notifications and its location also appeared.

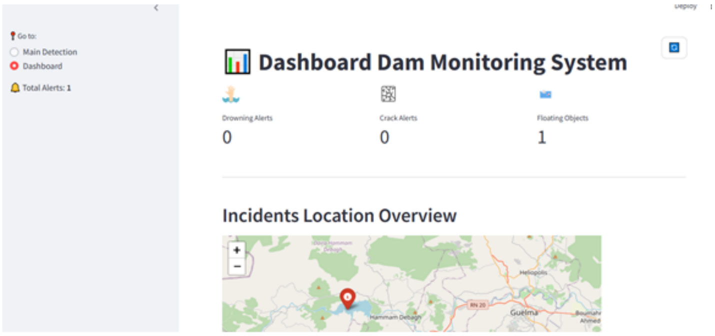


FIGURE 4.14: Detection alert (objects).

Here we also put a picture of the drowning detection:

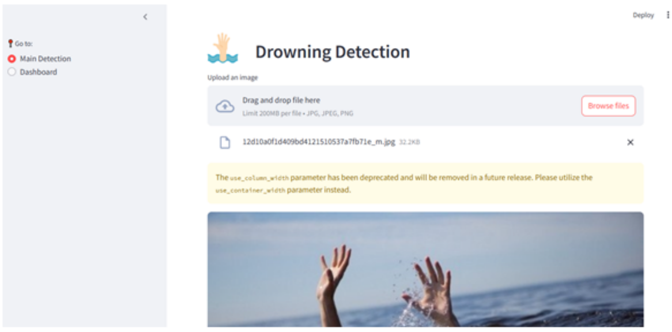


FIGURE 4.15: Detection Process (drowning).

We see that the system has also detected it and the number of alerts has increased:

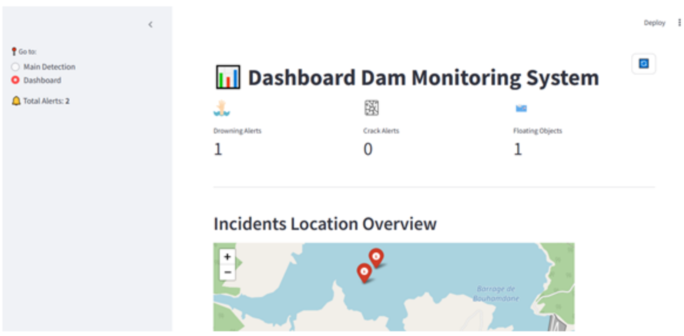


FIGURE 4.16: Detection alert (drowning).

This image shows the information card that appears after the examination:

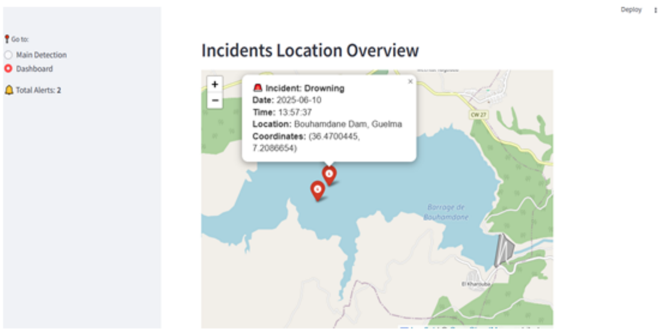


FIGURE 4.17: Information card that appears after the examination.

4.6 Conclusion

This chapter reviews the practical aspects of a smart dam monitoring system designed to detect three types of potential hazards: floating objects on the water surface, sinking conditions, and cracks in dam structures. Advanced computer vision and deep learning techniques were employed using YOLOv8 models after careful dataset preparation, including integration and data augmentation techniques.

The chapter begins with a detailed explanation of the training plan followed, identifying the settings and programming environments used to develop the models, whether on local computers or using cloud services such as Google Colab. Experimental performance results were then presented for each task, focusing on detection accuracy, response speed, and model performance in realistic scenarios simulating dam operating conditions.

The results demonstrated that the system is capable of providing accurate indicators of hazards, paving the way for its adoption as an aid in early monitoring and decision-making processes. This project highlights the great potential of employing artificial intelligence in developing smart solutions to protect critical infrastructure and represents an important step towards an automated and effective monitoring system that contributes to reducing risks and improving response in water resources management.

General conclusion

This project presents a practical and forward-looking approach to enhancing dam safety using artificial intelligence. A smart monitoring system was developed to address three critical challenges: detecting floating objects on the water surface that may obstruct dam operations, recognizing early signs of drowning incidents to enable timely intervention, and identifying structural cracks in dam walls as a preventive measure. The system relies on modern AI-based object detection techniquesYOLO, and was trained on carefully prepared and diverse datasets for each task. Specific attention was given to the complexity of drowning detection, which often involves subtle human movements, varying postures, and partial submersion—making it one of the most challenging scenarios to address. Thanks to robust data preparation, targeted augmentation, and careful class balancing, the system achieved strong performance across all tasks, even in visually complex or cluttered environments. One of the strengths of this solution is its ability to run on low-cost, resource-limited devices, making it highly applicable in remote or under-equipped environments. , making it suitable for real deployment in resource-constrained dam facilities. This project sets a solid foundation for future improvements, such as integrating real-time sensor data, testing in live dam environments, enhancing the user interface for rapid response, and expanding datasets to improve generalization. Ultimately, it demonstrates how AI can be used not only to monitor infrastructure but also to help save lives through early detection and timely action.

Chapter 5

Annex startup

5.1 Axe 01: Project Presentation

5.1.1 Project Idea

Our project involves developing an AI-powered smart system for dam monitoring and early hazard detection, focusing on safety and prevention. The idea emerged after conducting a field study and reviewing reports that revealed critical challenges facing dams in Algeria, such as the accumulation of floating objects, unobserved drowning incidents, and structural cracks. We are working on designing a scalable prototype of a smart platform that analyzes images captured by surveillance cameras installed around the dam. The system automatically detects floating objects, signs of drowning, and cracks in the structure using trained artificial intelligence models. Detected hazards are visualized on an interactive map, and immediate alerts are sent to the relevant authorities. The project is being developed by the Department of Computer Science as part of a university graduation project.

Avoiding Mechanical Failures and Downtime: Floating objects such as logs, debris, or boats can obstruct turbines and intake systems. Early detection helps maintain continuous operation and reduces emergency maintenance needs.

Enhancing Safety by Detecting Drowning Incidents: Real-time analysis of surveillance images enables quick detection of drowning cases, allowing immediate rescue actions and reducing the risk of fatalities.

Protecting the Environment and Ecosystem: Identifying floating waste helps prevent water pollution and supports ecological balance around the dam.

Instant Alerts with Geolocation: The system instantly notifies relevant teams of any detected hazard—whether structural, environmental, or human-related—along with its exact location on an interactive map.

Scalable and Cost-Effective Deployment: It utilizes existing camera infrastructure without requiring intrusive sensors, making it easy to implement across multiple dam locations.

User-Friendly Smart Interface: Operators can monitor threats through a simple web dashboard that displays alerts in real time, with no need for technical expertise.

5.1.2 Work Team

This project was designed and developed by Boumelit Yassamine, a master's student in Computer Science. She was responsible for creating the HydroGuard prototype, including the development of AI models, user interface design, and system integration. The work was carried out under the supervision and academic guidance of Mr. Khaled Halimi, His expertise and mentorship were essential in the orientation and success of the project.

5.1.3 Project Objectives

Short-Term Objectives

- Develop a smart solution for early detection of dam hazards like sinking, floating objects, and cracks.
- Validate feasibility by creating and testing a prototype with field partners.

- Establish initial partnerships with stakeholders in water and dam management.
- Increase awareness of dam risks and the solution's importance via campaigns and workshops.

Medium-Term Objectives

- Achieve adoption by local dam authorities as a monitoring and early warning tool.
- Expand into regional markets with similar infrastructure challenges.
- Enhance the interface and add features based on user feedback (e.g., risk reports, incident logs, alarm integration).

Long-Term Objectives

beginitemize

Become a leading reference in smart safety for dams and water infrastructure.

Extend the solution to sectors like ports, lakes, and drainage systems.

Help reduce disasters and financial losses caused by delayed hazard responses.

5.1.4 Implementation Schedule

Phases	**Month 1**	**Month 2**	**Month 3**	**Month 4**	**Month 5**	**Month 6**
Planning and preparation of a preliminary study	✓	✓				
Prototype development		✓	✓	✓		
Deployment and initial testing				✓	✓	
Testing and Optimization					✓	
Full Deployment						✓
Marketing and Promotion						✓

Table 5.1: Implementation Schedule

5.2 Axe 02: Innovative Aspects

5.2.1 Nature of the Innovations

AI-Powered Risk Detection: The project uses advanced AI and deep learning to monitor dam risks, detecting floating and sinking objects as well as tiny structural cracks. These models effectively recognize subtle image patterns beyond human or traditional system capabilities. **Real-Time Monitoring and Alerts**: The system analyzes images instantly and alerts officials immediately upon detecting hazards, enabling rapid intervention to prevent disasters like turbine blockages or crack worsening. **Interactive Risk Map Dashboard**: An interactive dashboard shows hazard locations on a smart map, with details such as detection time, hazard type (sinking, floating object, crack), and images—helping operators make quick, informed decisions without technical expertise. **Scalability and Flexibility**: Designed for expansion beyond adding cameras, the scalable prototype supports multiple dam sites locally and nationally. It adapts to varying conditions, transitions from recorded to live video analysis, and integrates smoothly with existing systems, allowing gradual adoption and ongoing development to meet safety needs.

5.2.2 Fields of Innovation

This project combines several innovative fields to create an advanced dam safety monitoring solution. Using AI and computer vision, it analyzes images to detect unusual objects or events in real time. The user interface features an interactive map that helps non-technical users make informed decisions. The system supports early warning and environmental risk management, enabling rapid responses to prevent damage or loss. Additionally, it offers future potential for expansion through multi-sensor integration and task diversification, evolving into a comprehensive tool for water infrastructure monitoring.

5.3 Axe 03: Strategic Market Analysis

5.3.1 Market Segment

- The primary market includes Algerian public bodies managing dams and water resources, especially the National Agency for Dams and Transfers (ANBT), responsible for over 80 dams.
- The Ministry of Water Resources (MMRE) plays a key role in national adoption and support.
- The National Sanitation Office (ONA) and National Office of Irrigation and Drainage (ONID) monitor facilities linked to reservoirs and canals.
- The General Directorate of Civil Protection relies on the system for flood response and accurate alerts.
- Engineering and supervision firms like CTE and SGI Algeria seek to integrate smart monitoring tools.
- Local water distribution agencies (AUEA) use the system to detect faults and leaks at municipal or regional levels.
- Private hotels, resorts, water parks, and pools represent potential clients needing continuous water safety monitoring.
- Fish farming operations can benefit in pond monitoring, foreign object detection, and water quality control.
- Seawater desalination plants gain from early detection of debris or leaks, enhancing preventive maintenance and efficiency.

5.3.2 Measuring the Intensity of Competition

Although our smart platform innovatively combines AI and image analysis to detect floating objects, sinking incidents, and cracks in dams, it enters a local market still dominated by traditional tools and intervention methods. Competition varies by platform function and falls into four main categories:

1. **Traditional Methods (Indirect Competition Through Manual and Visual Monitoring)**: Field actors rely on unintelligent methods for detection and intervention, especially in urgent or recurring cases.

    - In the case of floating objects and sinking cases, they rely on guards or citizens, and on surveillance cameras without any automatic analysis.
    - As for cracks, they are visually monitored during periodic field visits by technical teams.
    - *Weaknesses*:
        - Difficulty detecting small or rapidly moving objects.
        - Delayed reporting and intervention due to the absence of automatic alerts.
        - No prior behavioral analysis of sinking cases.
        - Lack of digital documentation and data accumulation over time.
        - Reliance on human judgment in assessing cracks, which reduces accuracy.

2. **Consulting Offices and Contractors – Traditional Partial Solutions**: Organizations like CTE Algérie, SGI Algérie, and BEAH handle dam monitoring and technical assessments, mainly for public institutions, relying on periodic field inspections and reports.

3. *Weaknesses*:

    - They do not provide instant solutions or continuous automated monitoring.
    - A complete lack of artificial intelligence tools or image analysis.
    - They provide partial services, usually limited to cracks without integration with other elements (sinking, floating objects).
    - Their reports are delayed, which reduces the effectiveness of rapid response.

Public Institutions (Target Market and Partial Competitors): Includes ANBT, Algerian Waters (ADE), and State Irrigation Directorates (DRE/DTP), which oversee dam supervision, monitoring, and periodic maintenance

Despite their pivotal role, they often:

- Rely on human field teams or external consulting firms.
- Lack intelligent systems capable of real-time analysis or alerts.
- Use surveillance cameras for documentation purposes only without integrating them with advanced analytical tools.

Weaknesses:

- Slow response to sudden events.
- Poor integration between agencies and a multiplicity of methods.
- Lack of digitization and smart transformation despite their strategic importance.

However, these institutions are not considered direct technical competitors, but rather represent a key target market that could benefit from the platform as a decision-making support tool and proactive monitoring.

Future Threats (Foreign Solutions or Local Startup): With the expansion of the digital innovation and artificial intelligence market, new competitors may emerge in the future, either:

- From Algerian startups working in the field of smart infrastructure technologies.
- Or through foreign partnerships (French, German, Chinese, etc.) offering smart monitoring solutions or AI-enabled equipment.
- *Weaknesses*:
  - There is currently no Algerian platform offering a comprehensive solution that integrates crack, floating, and sinking detection.
  - Awareness of smart solutions in this field remains limited.
  - The absence of competitors with strong local implementation capabilities opens the door to leadership.
- *Conclusion*: Despite the presence of traditional practices and some entities that may be considered partial competitors, our platform has a strong competitive position thanks to:
  - Its unique integration of three vital functions.
  - Its reliance on artificial intelligence and real-time image analysis.
  - Its provision of a reliable and effective early warning system.
- This makes it a leading opportunity in the Algerian market, both as a standalone project and as a solution that can be adopted and integrated by existing players.

5.3.3 Marketing Strategy

A multi-faceted marketing strategy has been developed to ensure widespread adoption of the system and tangible impact in the Algerian market, considering the specific stakeholders in dam and water safety sectors.

1. **Direct Sales**: Target decision-makers in public institutions (ANBT, ADE, DTP), engineering firms (SGI, CTE), and contractors through demos and technical documentation showcasing the system's value in prevention, efficiency, and cost savings.

2. **Strategic Partnerships**: Collaborate with inspection and engineering companies to integrate the system into projects, partner with university research centers for improvement and testing, and coordinate with Civil Protection and environmental agencies to broaden adoption.

3. **Free Trials and Incentives**: Offer a free trial at a pilot dam in cooperation with public institutions, provide a 20

4. **Awareness Campaigns and National Initiatives**: Link the project to national environmental safety and drowning prevention initiatives through participation in awareness events and campaigns like the "Clean Dam" initiative, with interactive field demos alongside local authorities.

5. **Digital and Media Presence**: Develop a simple website highlighting system features, user testimonials, and live demos; promote via Facebook and LinkedIn targeting technical agencies; produce short awareness videos in collaboration with journalists to present the project as an innovative local initiative.

5.4 Axe 04: Production and Organization Plan

5.4.1 Production Process

1. **Needs Study and Market Analysis**: Conducted a field study with dam management, maintenance engineers, civil protection, and ADE to identify real needs and early hazard detection challenges.

2. **System Design and Technical Architecture**: Developed a preliminary system design featuring back-end data collection and analysis, plus a front-end interface for alerts, designed flexibly for future expansion.

3. **Smart Model Development**: Created three AI models, each specialized in detecting floating objects, sinkholes, and cracks, trained on realistic datasets.

4. **Performance Testing and Optimization**: Performed tests in semi-realistic environments using simulated dam data; analyzed results with metrics like mAP to enhance model accuracy.

5. **Pilot Deployment and Integration**: Deployed a pilot platform with an interactive map interface, tested by water resource users to gather feedback and guide improvements.

6. **Follow-up and Maintenance**: Monitored system performance regularly post-launch, provided updates, and prepared a technical user guide to facilitate adoption by non-technical users.

5.4.2 Procurement

1. **Hardware**: Digital surveillance cameras (IP/RTSP), a GPU-equipped computer, and a GPS receiver/module for capturing precise incident coordinates.

2. **Software**:

- Geographic data management tools (e.g., Leaflet.js, Google Maps API) to display incident locations on an interactive map with exact GPS data.
- Alert systems (e.g., Firebase Notifications, Email APIs) to promptly notify dam authorities or rescue teams of critical events like floods or large cracks.
- Database systems (PostgreSQL, Firebase) to store incident records, images, coordinates, and alert details.

3. **Integration and Interface**:

    - UI and development frameworks (e.g., Flask, React, FastAPI) for a centralized platform allowing supervisors to monitor incidents and access detailed reports.
    - Incident management tools (dashboards, cards) presenting alert details: time, type, images, coordinates, and maps.

5.4.3 Workforce

Effective workforce organization is crucial for the smooth development of the AI-based smart dam monitoring platform. Due to the project's technical and multidisciplinary nature, the team must encompass diverse skills while maintaining a streamlined structure during initial phases. The primary roles include:

- Software Engineer: Develops the system and integrates components.
- AI Engineer: Trains and optimizes AI models.
- Network Technician: Configures cameras and communication interfaces.
- Project Manager: Oversees progress, coordinates the team, prepares reports, and ensures timeline adherence.

- Legal Advisor: Ensures compliance with technical and legal standards, drafts contracts, protects intellectual property, and provides legal counsel.
- UI/UX Designers: Design user-friendly and visually appealing interfaces.
- Marketing Specialists: Prepare promotional content and participate in events to present the platform.
- Customer/User Support Specialist: Provides technical assistance and resolves user inquiries to ensure effective system use.

Table 5.2: Team Structure and Number of Positions

Position	**Number**
Project Manager	1
Artificial Intelligence Engineers	1
Software Engineer	1
Network Technicians	1
Legal Consultant	1
UI/UX Designer	1
Marketing Specialist	1
User Support Specialis(1)	1

5.4.4 Key Partners

The success of the HydroGuard system relies on strategic collaborations with several key partners across technical, academic, and financial domains:

1. **Technical and Development Partners**:

   - Hardware Manufacturers: Companies providing high-quality smart surveillance cameras (e.g., Reolink) and GPU-based servers for real-time image processing.
   - Software and AI Partners: Platforms and experts in AI and machine learning (e.g., Roboflow, Google Colab, GitHub, Kaggle) contributing to the development and optimization of detection algorithms.

2. **Academic and Incubation Support**:

   - University Business Incubators: Offer pre-incubation support, mentorship, networking, and assistance with startup certification and funding access.

3. **Financial and Investment Partners**:

   - ALCOR Invest: Provides equity financing for innovative technology projects in Algeria.
   - ACF (Algeria Corporate Fund): A national public fund that supports high-potential startups, especially in AI, digital transformation, and energy.
   - Algeria Startup Fund: Official financial support for projects certified as Startups.
   - MDI (Mediterranean Development Initiative): Offers financial backing to entrepreneurial projects, often with international collaboration.
   - FINADEV Algérie: Microfinance and soft loans for small and emerging enterprises.
   - Algerian Business Angels Network (ABAN): A national network of angel investors focused on early-stage startups.
   - Algerian Startup Initiative (ASI): Offers mentorship, resources, and funding for tech-driven startups.

5.5 Axe 05: Financial Plan

5.5.1 Costs and Charges

1. **Technical Development Costs**:

- Developing AI algorithms specific to each function (detecting drowning, floating objects, and cracks).
- Building an interactive user interface and back-end system for image processing.
- Using specialized tools such as Roboflow and cloud training services (Google Colab, GPU servers).
- Testing models and continuously improving their performance.

2. **Infrastructure and Equipment Costs**:
    - Hosting the platform on secure and scalable cloud servers.
    - Powerful graphics processing units (GPUs) for training and operation.
    - High-resolution cameras for collecting field imagery.
3. **Human Resources Costs**:
    - Includes costs associated with staff and legal counsel costs.
4. **Training and Promotion Costs**:
    - Organizing training workshops for end users.
    - Designing promotional materials such as videos and a website.
    - Participating in innovation events or giving presentations to official bodies.
    - Building a visual identity and professional logo for the platform.
5. **Operating Costs**:
    - Operating expenses such as transportation, and communication means.

Detail	**Unit**	**Total**
Reolink Go PT Plus(4)	40.000.00	160.000.00
Development Tools(google Colab GitHub kaggle ...)	23.000.00	23.000.00
Sponsor Ads	20.000.00	20.000.00
Mobile phone line	12.000.00	12.000.00
transportation	25.000,00	25.000,00
security	25.000,00	25.000,00
Cloud hosting	15.000.00	15.000.00
GPU servers	40.000.00	40.000.00
Total	20.000.00	32.000.00

TABLE 5.3: Estimated Operational and Equipment Costs

Position	**Unit salary/Month**	**total salary/Month**
Project Manager(1)	80.000.00	80.000.00
Artificial Intelligence Engineers (1)	70.000.00	70.000.00
Software Engineer(1)	40.000.00	40.000.00
Network Technicians(2)	60.000.00	120.000.00
Legal Consultant (1)	(per task) 50.000.00	50.000.00
UI/UX Designer(1)	50.000.00	50.000.00
Marketing Specialist(1)	65.000.00	65.000.00
User Support Specialis(1)	55.000.00	55.000.00
Total	470,000.00	530,000.00

TABLE 5.4: Estimated Monthly Human Resource Costs

Position	**Monthly Salary (DA)**	**Duration (Months)**	**Total Cost (DA)**
Project Manager	80.000.00	6	480.000.00
AI Engineer	70.000.00	3 (Phase 2 to 4)	210.000.00
Software Engineer	40.000.00	3 (Phase 2 to 4)	120.000.00
Network Technicians (x2)	120.000.00	2 (Phase 5 to 6)	240.000.00
Legal Consultant	50.000.00 (per task)	1	50.000.00
UI/UX Designer	50.000.00	1 (Phase 2)	50.000.00
Marketing Specialist	65.000.00	1 (Phase 6)	65.000.00
User Support Specialis	55.000.00	1	55.000.00
Total			1.270.000.00

TABLE 5.5: Estimated Monthly Human Resource Costs

5.5.2 Pricing

Now, to set a price for the system, we perform the following calculation: Technical costs + human resources costs + sales margin = 32.000.00 + 1.270.000.00 + 30%

sales margin = 1,590,000 × 0.30 = 477,000 DA

So: 1,590,000 + 477,000 = 2,067,000 DA

From this we conclude that the final price is: 2,067,000 DA

Final Selling Price = 2,067,000 DA

- Since our system focuses on three points and we have potential markets whose interest in one part may have prompted us to sell the system as follows:

Component	**Proposed Fixed Price**
Drowning Detection	689,000 DA
Floating Object Detection	689,000 DA
Crack Detection	689,000 DA
Full System	2,067,000 DA

Table 5.6: Breakdown of Fixed Prices per Detection Module

5.5.3 Costs and Charges

Based on a national market study, multi-scenario financial forecasts (conservative and optimistic) were prepared to estimate turnover over a three-year period. These forecasts are based on incremental market penetration assumptions, taking into account the possibility of selling the system in full or in part, depending on customer needs (drowning detection, crack detection, or floating object detection).

Component	**The first year**	**The second year**	**The third year**
Full System	1	2	3
Drowning Detection	2	3	6
Crack Detection	0	1	2
Floating Object Detection	1	2	3

Table 5.7: Pessimistic Revenue Projection

Year 1 Revenue: 4,134,000 DA Year 2 Revenue: 8,268,000 DA Year 3 Revenue: 13,780,000 DA

Component	The first year	The second year	The third year
Full System	3	6	3
Drowning Detection	4	8	6
Crack Detection	1	3	4
Floating Object Detection	2	3	4

Table 5.8: Optimistic Revenue Projection

Year 1 Revenue: 11,024,000 DZD Year 2 Revenue: 21, 492,000 DZD Year 3 Revenue: 15,847,000 DZD

Year	Year Turnover (DA)	Operating Costs (DA)	Net Profit (DA)
First year	4,134,000	1,600,000	2,534,000
Second year	8,268,000	1,600,000	6,668,000
Third year	13,780,000	1,600,000	6,668,000

Table 5.9: Financial Analysis (pessimist)

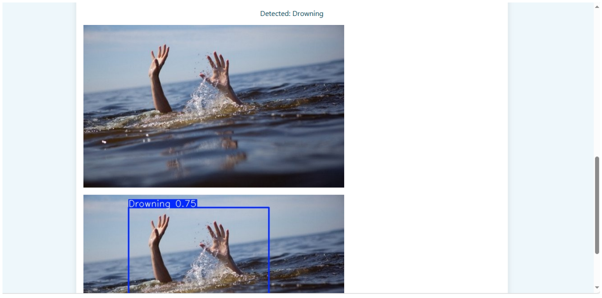
Year	Year Turnover (DA)	Operating Costs (DA)	Net Profit (DA)
First year	11,024,000	1,600,000	9,424,000
Second year	21,492,000	1,600,000	19,892,000
Third year	15,847,000	1,600,000	14,247,000

Table 5.10: Optimistic Scenario Financial Analysis

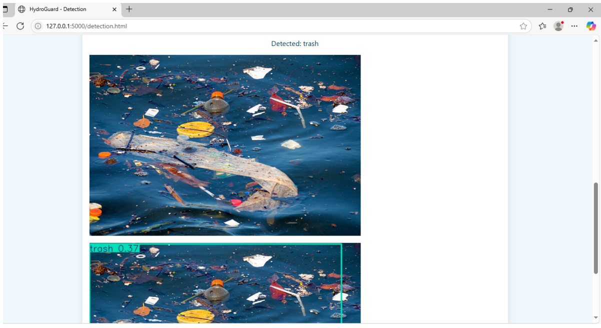
5.6 Axe 06: Experimental Prototype

5.6.1 Prototype

The following images show the experimentation of the three tasks:



(A) Drowning Detection

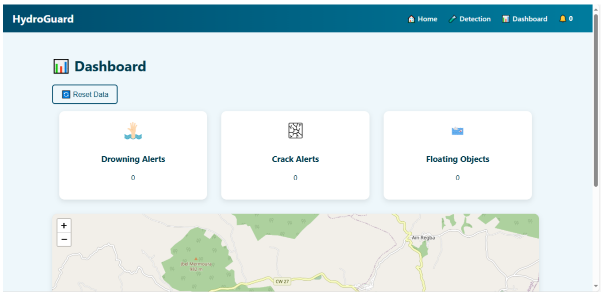


(B) Floating Objects Detection

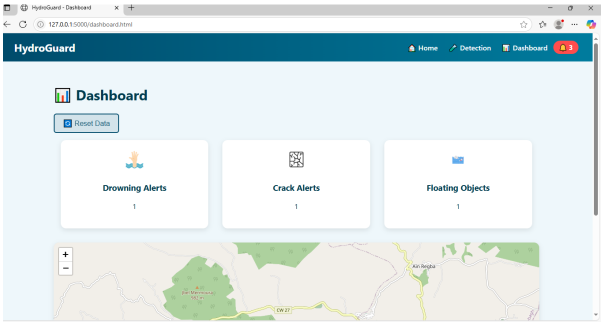
(C) Crack Detection

FIGURE 5.1: Overview of Detection Modules

They also show how the number of alerts changed from zero to three.



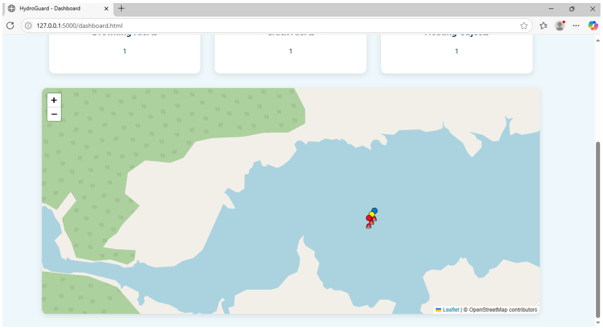
(A) Dashboard before detecting any alert



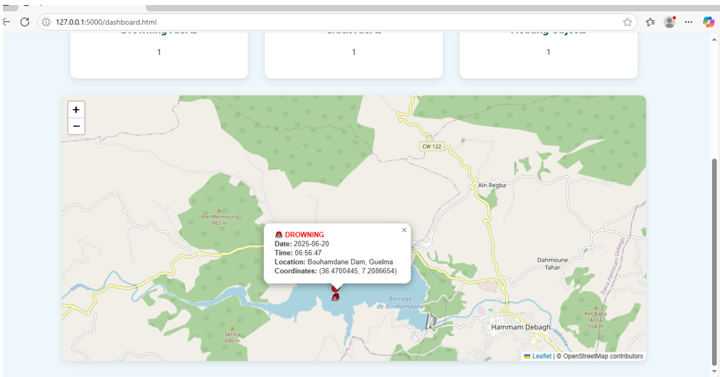
(B) Dashboard after detecting a hazard

FIGURE 5.2: Comparison of dashboard status before and after hazard detection

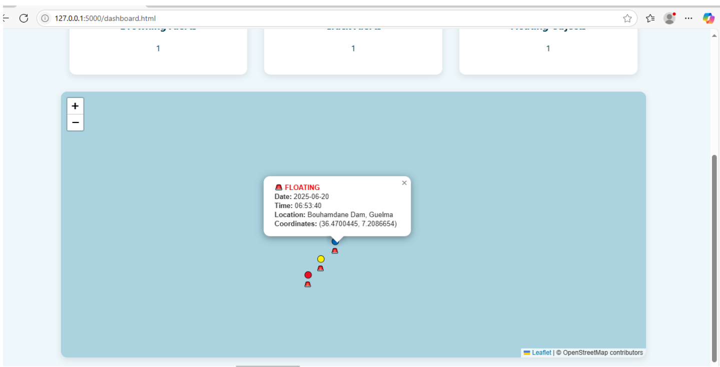
These pictures show how the site appears with information on the map



(A) Alert displayed on the map



(B) Example of a drowning incident – information displayed



(C) Example of a floating object detection – details displayed

FIGURE 5.3: Dashboard interface showcasing alert, drowning case, and floating object detection

This is the website link: https://it.univ-guelma.dz/HydroGuard/

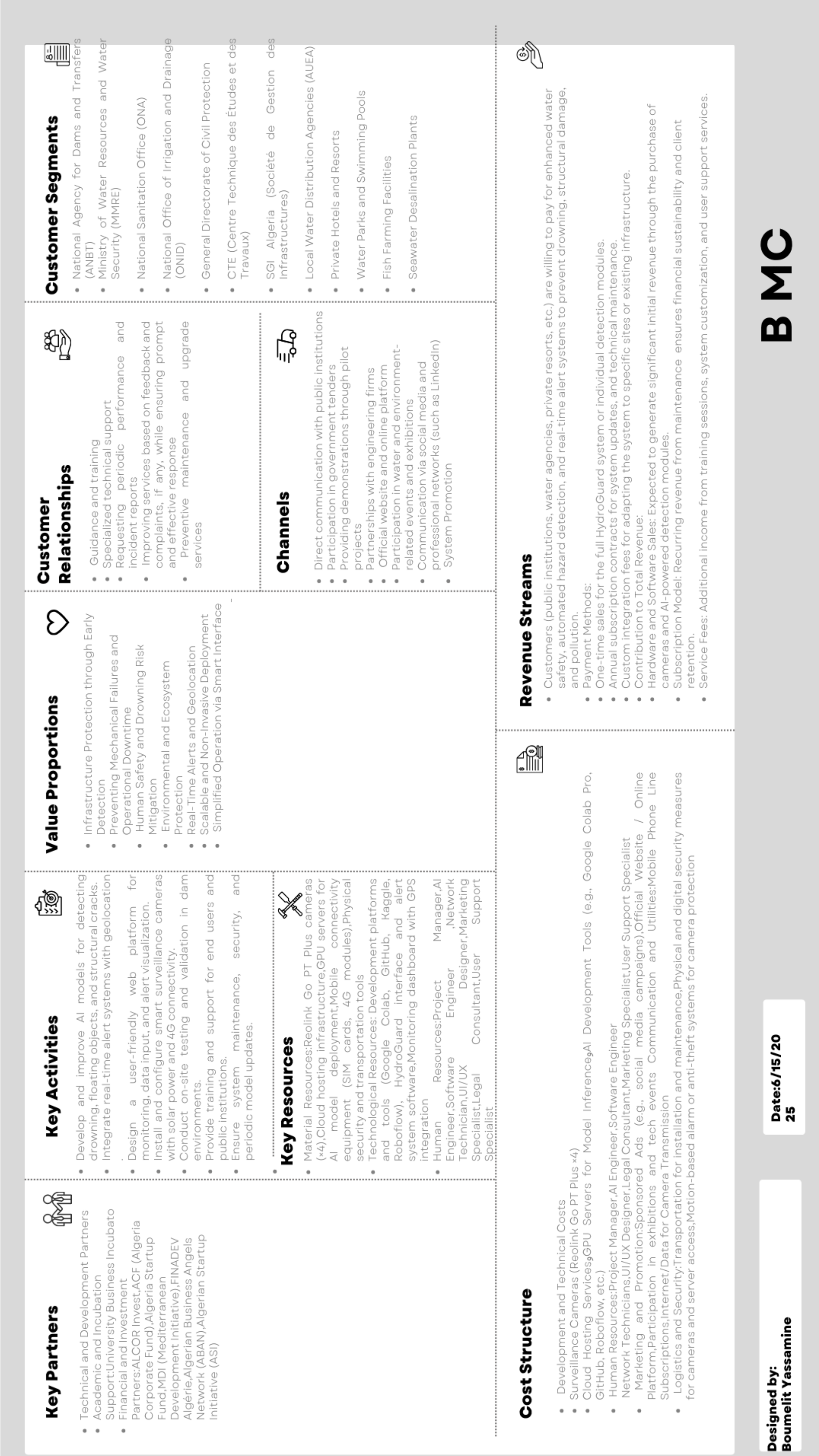


FIGURE 5.4: BMC

Bibliography

[1] Food and Agriculture Organization of the United Nations (FAO). *AQUASTAT Database*. `http://www.fao.org/aquastat/`. Accessed: January 4, 2025. 2025.

[2] J. Chen **andothers**. **?**Population, Water, Food, Energy and Dams**? in***Renewable and Sustainable Energy Reviews*: 56 (2016), **pages** 18–28. DOI: `10.1016/j.rser.2015.11.018`.

[3] Doğan Altinbilek. **?**The role of dams in development**? in***Water Science and Technology*: 45.8 (2002), **pages** 169–180.

[4] Mengxia Zhao, Yanyi Liu, Ying Wang **andothers**. **?**Effectiveness assessment of reservoir projects for flash flood control, water supply and irrigation in Wangmo Basin, China**? in***Science of the Total Environment*: 851 (2022), **page** 157911. DOI: `10.1016/j.scitotenv.2022.157911`.

[5] Sakshi Naithani **and** Ashis Kumar Saha. **?**Changing landscape and ecotourism development in a large dam site: a case study of Tehri dam, India**? in***Asia Pacific Journal of Tourism Research*: 24.3 (2019), **pages** 193–205. DOI: `10.1080/10941665.2018.1561473`.

[6] J. A. Dixon, L. M. Talbot **and** G. J.-M. Le Moigne. *Dams and the Environment: Considerations in World Bank Projects*. `https://documents1.worldbank.org/curated/zh/300561468782146320/pdf/multi-page.pdf`. Accessed: January 22, 2025. 1989.

[7] Revistas Electrónicas UJAEN. *Article sur la conservation des ouvrages.* `https://revistaselectronicas.ujaen.es/index.php/atma/article/download/5298/6334?inline=1`. Accessed: January 22, 2025. n.d.

[8] Boualem Remini **andothers**. *Situation géographique des barrages les plus menacés par l'envasement.* Accessed: January 22, 2025. 2020. URL: `https://www.researchgate.net/figure/Situation-geographique-des-barrages-les-plus-menaces-par-lenvasement-Remini-et_fig2_343636574`.

[9] Boualem Remini. *Situation géographique des barrages les plus menacés par l'envasement.* Accessed: January 21, 2025. 2020. URL: `https://www.researchgate.net/figure/Situation-geographique-des-barrages-les-plus-menaces-par-lenvasement-Remini-et_fig2_343636574`.

[10] Abdelkader Hadidi **andothers**. **?**Study of the relationship between the siltation rate of Algerian dams and the runoff coefficient**? in***H2Open Journal*: 6.4 (2023), **pages** 535–550.

[11] Agence Nationale des Barrages et Transferts (ANBT). *Site officiel de l'ANBT.* Accessed: January 26, 2025. 2025. URL: `http://www.anbt.dz`.

[12] Kawther Touhami, Ramdane Chihaoui, Abdelhamid Mejdoub Nefoussi **andothers**. **?**Investigation of the potential use of dredged dam sediments as supplementary cementitious materials**? in***Construction and Building Materials*: 422 (2024), **page** 135730.

[13] Omar Djoukbala, Salim Djerbouai, Saeed Alqadhi **andothers**. **?**A geospatial approach-based assessment of soil erosion impacts on the dams silting in the semi-arid region**? in***Geomatics, Natural Hazards and Risk*: 15.1 (2024), **page** 2375543.

[14] Nasrat Adamo, Nadhir Al-Ansari, Varoujan Sissakian **andothers**. **?**Dam safety: sediments and debris problems**? in***Journal of Earth Sciences and Geotechnical Engineering*: 11.1 (2021), **pages** 27–63.

[15] Elaid Madene, Abdelmadjid Boufekane, Gianluigi Busico **andothers**. **?**Temporal evaluation of surface water quality in semiarid regions: a case study of the Ouled Mellouk dam in north-western Algeria**? in** *Water Supply*: 23.6 (2023), **pages** 2387–2403.

[16] APS (Algerian Press Service). *80 drowning cases recorded in dams since the beginning of the summer season*. Accessed: May 21, 2025. 2023. URL: `https://www.aps.dz/ar/societe/147383-2023-07-26-13-37-53`.

[17] APS (Algerian Press Service). *Civil Protection records 76 drowning cases in dams during the summer season*. Accessed: May 21, 2025. 2023. URL: `https://www.aps.dz/ar/societe/93366-76-4`.

[18] Yves-François Le Lay **and** Bertrand Moulin. **?**Les barrages face à la problématique des bois flottants: collecte, traitement et valorisation**? in***La Houille Blanche*: 3 (2007), **pages** 96–103.

[19] Association of State Dam Safety Officials. *Reservoir Debris Management - Lessons Learned*. techreport. Final Report No. 4781. National Reservoir Sedimentation **and** Debris Management Program, 2022.

[20] Nasrat Adamo, Nadhir Al-Ansari, Varoujan Sissakian **andothers**. **?**Dam safety: sediments and debris problems**? in***Journal of Earth Sciences and Geotechnical Engineering*: 11.1 (2021), **pages** 27–63.

[21] Mohammed Ibrahim Bessaid. **?**Contribution à l'étude du comportement sismique des barrages en béton**?** Thèse de doctorat en Génie Civil, spécialité Génie Parasismique. phdthesis. Université Abou Bekr Belkaïd – Tlemcen, 2022. DOI: `10.13140/RG.2.2.28956.46727`.

[22] Jason W. Poff **and** Rollin H. Hotchkiss. **?**Stream slope as an indicator for drowning potential at low head dams**? in** *Water*: 15.3 (2023), **page** 512.

[23] Yongeun Park, SungHwan Kim, Kangmin Chon **andothers**. ?Impacts of heavy rain and floodwater on floating debris entering an artificial lake (Daecheong Reservoir, Korea) during the summer? **in***Desalination and Water Treatment*: 219 (2021), **pages** 399–404.

[24] Zhang Han, Yanling Li, Zepeng Zhao **andothers**. ?An Online safety monitoring system of hydropower station based on expert system? **in***Energy Reports*: 8 (2022), **pages** 1552–1567.

[25] Qi Song **andothers**. ?MS-YOLO: A Lightweight and High-Precision YOLO Model for Drowning Detection? **in***Sensors*: 24.21 (2024), **page** 6955.

[26] Guru Prakash, Revanth Dugalam, Mohamed Barbosh **andothers**. ?Recent advancement of concrete dam health monitoring technology: A systematic literature review? **in***Structures*: Elsevier, 2022, **pages** 766–784.

[27] Ling Zhu, Zhenhong Li, Xiaolei Huang **andothers**. ?Advances in geodetic techniques for monitoring crustal deformation and dam displacements: A review? **in***Sensors*: 18.11 (2018), **page** 3682.

[28] Vinay Kumar, Mahesh Kumar Singh, Anand Sharma **andothers**. ?IoT based Smart Dam Monitoring System? **in***IJARCCE*: 8.6 (2019), **pages** 15–21.

[29] Jingyu Zhang, Ao Xiang, Yu Cheng **andothers**. ?Research on detection of floating objects in river and lake based on AI intelligent image recognition? **in***arXiv preprint arXiv:2404.06883*: (2024).

[30] Pradyumna Gokhale, Omkar Bhat **and** Sagar Bhat. ?Introduction to IOT? **in***International Advanced Research Journal in Science, Engineering and Technology*: 5.1 (2018), **pages** 41–44.

[31] *Smart Dams: Integrating IoT and AI for Dam*. 2024.

[32] *Smart Dams: Integrating IoT and AI for Dam Monitoring*. `https://www.ijcet.org`. Accessed: January 25, 2025. 2023.

[33] *IoT and Smart Systems for Structural Health Monitoring of Dams.* `https://www.mdpi.com/journal/sensors`. Accessed: January 25, 2025. 2024.

[34] ?Application of LiDAR for flood and dam monitoring? **in***Remote Sensing of Environment*: (2018).

[35] ?Radar-based monitoring in water bodies? **in***IEEE Transactions on Geoscience and Remote Sensing*: (2021).

[36] ?A review of ultrasonic level sensors in water management? **in***Journal of Environmental Monitoring and Assessment*: (2015).

[37] ?Radar sensors for monitoring water levels? **in***Sensors and Actuators A: Physical*: (2019).

[38] ?Infrared temperature sensors for remote water surface monitoring? **in***Sensors and Instrumentation, Journal of Water Resources*: (2018).

[39] ?The role of humidity sensors in environmental monitoring? **in***Environmental Monitoring and Assessment*: (2017).

[40] ?Pressure sensors in dam safety monitoring? **in***Journal of Civil Engineering and Environmental Systems*: (2020).

[41] Orange. *Tout savoir sur la connectivité LoRa. IoT Journey.* Available at: `https://iotjourney.orange.com/fr-FR/connectivite/tout-savoir-sur-la-connectivite-lora`, Accessed: January 23, 2025.

[42] StudySmarter. *Zigbee.* Available at: `https://www.studysmarter.fr/resumes/ingenierie/ingenierie-des-telecommunications/zigbee/`, Accessed: June 14, 2025.

[43] Eskuad. *How Cloud-Based Data Storage Enhances Security and Accessibility.* Available at: `https://blog.eskuad.com/how-cloud-based-data-storage-enhances-security-and-accessibility`, Accessed: January 23, 2025.

[44] Cleaq. *Cloud vs stockage local : Quelle solution choisir pour son entreprise ?* Available at: https://www.cleaq.com/blog/equipements-informatiques/cloud-vs-stockage-local-quelle-solution-choisir-pour-son-entreprise, Accessed: anuary 26, 2025.

[45] European Commission. *White Paper on Artificial Intelligence: A European approach to excellence and trust*. 2020.

[46] IEEE. *IEEE Standard for Artificial Intelligence Ethics and Governance*. IEEE.

[47] Co-Learning Lounge. *What is Deep Learning AI? (In Simple Words)*. https://medium.com/co-learning-lounge/what-is-deep-learning-ai-in-simple-words-ad2c39e13bf2. Accessed: january 23 , 2025. n.d.

[48] ResearchGate. *Types of machine learning: Machine learning encompasses three main types—supervised, unsupervised, and reinforcement learning*. https://www.researchgate.net/figure/Types-of-machine-learning-Machine-learning-encompasses-three-main-types-supervised_fig1_373535780. Accessed: June 14, 2025. n.d.

[49] International Organization for Standardization (ISO). *ISO/IEC 22989:2022 - Artificial Intelligence — Concepts and Terminology*. Standard published by ISO. 2022.

[50] Stuart J. Russell **and** Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2016.

[51] Abderrahim Sebri. *Cours Machine Learning (ML) & Intelligence Artificielle (IA)*. Cours personnel ou support pédagogique. 2022.

[52] ResearchGate. *The relationship between machine learning and human learning*. Accessed: January 26, 2025. 2025. URL: https://www.researchgate.net/figure/The-relationship-between-machine-learning-and-human-learning_fig3_376616162.

[53] iArtificial Blog. *Différence entre l'apprentissage automatique et l'intelligence artificielle*. Accessed: January, 2025. 2022.

[54] *Human Learning and Machine Learning: How They Differ*. Accessed: January, 2025. 2025. URL: `https://www.datasciencecentral.com/human-learning-and-machine-learning-how-they-differ/`.

[55] NashTech Global. *Architecture of Artificial Neural Network*. Accessed: January, 2025. 2025. URL: `https://blog.nashtechglobal.com/architecture-of-artificial-neural-network/`.

[56] Nirodya Pussadeniya. *Introduction to CNN Architecture*. Accessed: January, 2025. 2025. URL: `https://medium.com/@Nirodya_Pussadeniya/introduction-to-cnn-architecture-53f2f8e8705f`.

[57] Keiron O'Shea **and** Ryan Nash. **?**An introduction to convolutional neural networks**? in***arXiv preprint arXiv:1511.08458*: (2015).

[58] Qingsen Wu **andothers**. **?**Deep neural networks-based direct-current operation prediction and circuit migration design**? in***Electronics*: 12.13 (2023).

[59] Raphael Kassel. *Deep Neural Network : qu'est-ce qu'un réseau de neurones profond ?* Accessed: February, 2025. DataScientest. 06 **april** 2021. URL: `https://datascientest.com/deep-neural-network`.

[60] Adem R.N. Aouichaoui **andothers**. **?**Comparison of group-contribution and machine learning-based property prediction models with uncertainty quantification**? in***31st European Symposium on Computer Aided Process Engineering*: **byeditor**Metin Türkay **and** Rafiqul Gani. **volume** 50. Computer Aided Chemical Engineering. Elsevier, 2021, **pages** 755–760.

[61] Robin M. Schmidt. **?**Recurrent neural networks (RNNs): A gentle introduction and overview**? in***arXiv preprint arXiv:1912.05911*: (2019).

[62] GeeksforGeeks. *Deep Learning: Introduction to Long Short-Term Memory (LSTM)*. Consulté le : 22 février 2025. 2024. URL: `https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/`.

[63] Zhaoqing Pan, Weijie Yu, Xiaokai Yi **andothers**. ?Recent progress on generative adversarial networks (GANs): A survey? **in***IEEE Access*: 7 (2019), **pages** 36322–36333.

[64] Maxim Lapan. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd, 2018.

[65] IBM. *Transfer Learning: What it is and how it works*. Accessed: February 28, 2025. 2025. URL: `https://www.ibm.com/think/topics/transfer-learning`.

[66] Zhaoyang Niu, Guoqiang Zhong **and** Hui Yu. ?A review on the attention mechanism of deep learning? **in***Neurocomputing*: 452 (2021), **pages** 48–62.

[67] Ozan İrsoy **and** Ethem Alpaydın. ?Unsupervised feature extraction with autoencoder trees? **in***Neurocomputing*: 258 (2017). Special Issue on Machine Learning, **pages** 63–73.

[68] Görkem Algan **and** Ilkay Ulusoy. ?Image classification with deep learning in the presence of noisy labels: A survey? **in***Knowledge-Based Systems*: 215 (2021), **page** 106771.

[69] Kaggle Community. *Getting Started with Kaggle: Discussion Thread*. Accessed: April 11, 2025. 2024. URL: `https://www.kaggle.com/discussions/getting-started/463662`.

[70] Roboflow Universe. *Marine Dataset O4WDF - Marine Object Detection*. Accessed: June 14, 2025. 2025. URL: `https://universe.roboflow.com/marine-object-detection/marine-dataset-o4wdf`.

[71] Roboflow Universe – XplodingDog. *DawgSurfaceCracksSAG Dataset*. Accessed: June 14, 2025. 2025. URL: `https://universe.roboflow.com/xplodingdog/dawgsurfacecrackssag`.

[72] Utah State University - Digital Commons. *All Datasets Collection, Dataset No. 48*. Accessed: June 14, 2025. 2025. URL: `https://digitalcommons.usu.edu/all_datasets/48/`.

[73] Roboflow Universe – Jimmy. *Drowning Detection IK28A Dataset*. Accessed: June 14, 2025. 2025. URL: `https://universe.roboflow.com/jimmy-vqsks/drowning-detection-ik28a`.

[74] Chang Che, Haotian Zheng, Zengyi Huang **andothers**. **?**Intelligent robotic control system based on computer vision technology**?** **in***arXiv preprint arXiv:2404.01116*: (2024).

[75] Hangcheng Dong, Nan Wang, Dongge Fu **andothers**. **?**Precision and Efficiency in Dam Crack Inspection: A Lightweight Object Detection Method Based on Joint Distillation for Unmanned Aerial Vehicles (UAVs)**?** **in***Drones*: 8.11 (2024), **page** 692.

[76] Guangchao Qiao, Mingxiang Yang **and** Hao Wang. **?**A detection approach for floating debris using ground images based on deep learning**?** **in***Remote Sensing*: 14.17 (2022), **page** 4161.

[77] He Li, Shuaipeng Yang, Rui Zhang **andothers**. **?**Detection of floating objects on water surface using YOLOv5s in an edge computing environment**?** **in***Water*: 16.1 (2023), **page** 86.

[78] Adel Mmar, Anis Koubaa, Mohanned Ahmed **andothers**. **?**Aerial images processing for car detection using convolutional neural networks: Comparison between faster r-cnn and yolov3**?** **in***arXiv preprint arXiv:1910.07234*: (2019).

[79] Tsung-Yi Lin **andothers**. **?**Focal loss for dense object detection**?** **in***CoRR*: abs/1708.02002 (2017).

[80] Joseph Redmon **andothers**. ?You only look once: Unified, real-time object detection? **in***CoRR*: abs/1506.02640 (2015).

[81] SZJY LED. *Frame Rate (FPS) vs. Refresh Rate (Hz): Do You Know Their Difference?* Accessed: 2025-06-25. 2024. URL: `https://www.szjy-led.com/frame-rate-vs-refresh-rate/`.

[82] Joseph Redmon **andothers**. ?You only look once: Unified, real-time object detection? **in***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: 2016, **pages** 779–788.

[83] Aishwarya Singh. *Selecting the right bounding box using non-max suppression (with implementation)*. Accessed: 2025-08-13. 2020. URL: `https://www.analyticsvidhya.com/blog/2020/08/selecting-the-right-bounding-box-using-non-max-suppression-with-implementation/`.

[84] Aishwarya Singh. *Selecting the Right Bounding Box Using Non-Max Suppression (with Implementation)*. Accessed: 2025-08-13. 2020. URL: `https://www.analyticsvidhya.com/blog/2020/08/selecting-the-right-bounding-box-using-non-max-suppression-with-implementation/`.

[85] Joseph Redmon **and** Ali Farhadi. ?YOLO9000: Better, faster, stronger? **in***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: 2017, **pages** 7263–7271.

[86] Joseph Redmon **and** Ali Farhadi. ?YOLO9000: Better, faster, stronger? **in***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: 2017, **pages** 7263–7271.

[87] Joseph Redmon **and** Ali Farhadi. ?Yolov3: An incremental improvement? **in***arXiv preprint*: arXiv:1804.02767 (2018).

[88] Alexey Bochkovskiy, Chien-Yao Wang **and** Hong-Yuan Mark Liao. ?Yolov4: Optimal speed and accuracy of object detection? **in***arXiv preprint*: arXiv:2004.10934 (2020).

[89] Ultralytics. *YOLOv5*. Accessed: May 5, 2025. 2025. URL: `https://docs.ultralytics.com/fr/models/yolov5/`.

[90] Chuyi Li **andothers**. ?Yolov6 v3.0: A full-scale reloading? **in***arXiv preprint*: arXiv:2301.05586 (2023).

[91] Chien-Yao Wang, Alexey Bochkovskiy **and** Hong-Yuan Mark Liao. ?YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors? **in***Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*: 2023, **pages** 7464–7475.

[92] M. Yaseen. ?What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector? **in***arXiv preprint*: arXiv:2408.15857 (2024).

[93] Fuxun Chen, Lanxin Zhang, Siyu Kang **andothers**. ?Soft-NMS-enabled YOLOv5 with SIOU for small water surface floater detection in UAV-captured images? **in***Sustainability*: 15.14 (2023), **page** 10751.

[94] Guangchao Qiao, Mingxiang Yang **and** Hao Wang. ?An annotated Dataset and Benchmark for Detecting Floating Debris in Inland Waters? **in***Scientific Data*: 12.1 (2025), **page** 385.

[95] Xiaohong Zhang, Changzhuo Min, Junwei Luo **andothers**. ?YOLOv5-FF: detecting floating objects on the surface of fresh water environments? **in***Applied Sciences*: 13.13 (2023), **page** 7367.

[96] Qi Song, Bodan Yao, Yunlong Xue **andothers**. ?MS-YOLO: A Lightweight and High-Precision YOLO Model for Drowning Detection? **in***Sensors*: 24.21 (2024), **page** 6955.

[97] He Li, Shuaipeng Yang, Rui Zhang **andothers**. ?Detection of floating objects on water surface using YOLOv5s in an edge computing environment? **in***Water*: 16.1 (2023), **page** 86.

[98] Jingyu Zhang, Ao Xiang, Yu Cheng **andothers**. **?**Research on detection of floating objects in river and lake based on AI intelligent image recognition**? in***arXiv preprint*: arXiv:2404.06883 (2024).

[99] Feng Lin, Tian Hou, Qiannan Jin **andothers**. **?**Improved YOLO based detection algorithm for floating debris in waterway**? in***Entropy*: 23.9 (2021), **page** 1111.

[100] Jie Wang **and** Hong Zhao. **?**Improved YOLOv8 Algorithm for Water Surface Object Detection**? in***Sensors*: 24.15 (2024), **page** 5059.

[101] Liangtian Zhao, Shouqiang Qiu **and** Yuanming Chen. **?**Enhanced Water Surface Object Detection with Dynamic Task-Aligned Sample Assignment and Attention Mechanisms**? in***Sensors*: 24.10 (2024), **page** 3104.

[102] Ruiliang Yang, Kaikai Wang **and** Libin Yang. **?**An improved YOLOv5 algorithm for drowning detection in the indoor swimming pool**? in***Applied Sciences*: 14.1 (2023), **page** 200.

[103] Nouf Alharbi, Layan Aljohani, Anhar Alqasir **andothers**. **?**Improved Automatic Drowning Detection Approach with YOLOv8**? in***Engineering, Technology & Applied Science Research*: 14.6 (2024), **pages** 18070–18076.

[104] World Scientific and Engineering Academy and Society. **?**WSEAS Transactions on Information Science and Applications**? in** *WSEAS*: (2025).

[105] Ruiliang Yang, Kaikai Wang **and** Libin Yang. **?**An improved YOLOv5 algorithm for drowning detection in the indoor swimming pool**? in***Applied Sciences*: 14.1 (2023), **page** 200.

[106] Nouf Lharbi **andothers**. **?**Improved Automatic Drowning Detection Approach with YOLOv8**? in***Engineering, Technology & Applied Science Research*: 14.6 (2024), **pages** 18070–18076.

[107] Xuwei Dong, Yang Liu **and** Jinpeng Dai. **?**Concrete Surface Crack Detection Algorithm Based on Improved YOLOv8**? in***Sensors*: 24.16 (2024), **page** 5252.

[108] Lasitha Piyathilaka, D. M. G. Preethichandra, Umer Izhar **andothers**. ?Real-time concrete crack detection and instance segmentation using deep transfer learning? **in***Engineering Proceedings*: **volume** 2. 1. 2020, **page** 91.

[109] Saeid Ataei, Saeed Adibnazari **and** Seyyed Taghi Ataei. ?Data-driven Detection and Evaluation of Damages in Concrete Structures: Using Deep Learning and Computer Vision? **in***arXiv preprint*: arXiv:2501.11836 (2025).

[110] Qinjun Li, Guoyu Zhang **and** Ping Yang. ?CL-YOLOv8: Crack Detection Algorithm for Fair-Faced Walls Based on Deep Learning? **in***Applied Sciences*: 14.20 (2024), **page** 9421.

[111] Haomin Zuo, Zhengyang Li, Jiangchuan Gong **andothers**. ?Intelligent road crack detection and analysis based on improved YOLOv8? **in***arXiv preprint*: arXiv:2504.13208 (2025).

[112] Md Abdur Rouf, Yuji Iwahori, Haisong Chen **andothers**. ?A novel approach for concrete crack and spall detection based on improved yolov8? **in***Proceedings of the 2023 5th International Conference on Video, Signal and Image Processing*: 2023, **pages** 149–155.

[113] Xuwei Dong, Yang Liu **and** Jinpeng Dai. ?Concrete Surface Crack Detection Algorithm Based on Improved YOLOv8? **in***Sensors*: 24.16 (2024), **page** 5252.

[114] Babloo Kumar **and** Sayantari Ghosh. ?Detection of concrete cracks using dual-channel deep convolutional network? **in***2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*: IEEE, 2020, **pages** 1–7.

[115] Linfeng Gan, Hu Liu, Yue Yan **andothers**. ?Bridge bottom crack detection and modeling based on faster R-CNN and BIM? **in***IET Image Processing*: 18.3 (2024), **pages** 664–677.

[116] Thai Son Tran, Son Dong Nguyen, Hyun Jong Lee **andothers**. ?Advanced crack detection and segmentation on bridge decks using deep learning? **in***Construction and Building Materials*: 400 (2023), **page** 132839.

[117] Range King. *Brief summary of yolov8 model structure.* Accessed: June 8, 2025. 2025. URL: `https://github.com/ultralytics/ultralytics/issues/189`.

[118] Pratik Kanani **and** Mamta Padole. **?**Deep learning to detect skin cancer using Google Colab**? in***International Journal of Engineering and Advanced Technology Regular Issue*: 8.6 (2019), **pages** 2176–2183.

[119] Python Team. *The Python Tutorial.* Accessed: 2025-05-8. 2024. URL: `https://docs.python.org/3/tutorial/index.html`.

[120] NVIDIA. *PyTorch.* Accessed: 2025-05-11. 2024. URL: `https://www.nvidia.com/en-us/glossary/pytorch/`.

[121] Domino Data Lab. *What is Jupyter Notebook?* Accessed: 2025-05-28. 2024. URL: `https://domino.ai/data-science-dictionary/jupyter-notebook`.

[122] Python Team. *The Python Tutorial.* Accessed: 2025-05-28. 2025. URL: `https://docs.python.org/3/tutorial/index.html`.

[123] NVIDIA. *PyTorch.* Accessed: 2025-05-28. 2025. URL: `https://www.nvidia.com/en-us/glossary/pytorch/`.

[124] Domino Data Lab. *What is Jupyter Notebook?* Accessed: 2025-05-28. 2025. URL: `https://domino.ai/data-science-dictionary/jupyter-notebook`.

[125] OpenCV. *Official Website.* Accessed: 2025-05-28. 2025. URL: `https://opencv.org/`.

[126] ActiveState. *What is Matplotlib in Python & How to Use it for Plotting?* Accessed: 2025-05-28. 2025. URL: `https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/`.