Politecnico di Torino

College of Electronic, Telecommunications and Physical Engineering

Master's Degree in ICT For Smart Societies

Master's Degree Thesis

Artificial Intelligence for Road Safety: Automated Detection, Assessment, and Validation of Highway Guardrails

Supervisor:
Prof. Marco PIRAS

Candidate:
Marika ATTANASIO

Academic year 2024-2025

Abstract

This thesis explores the application of Artificial Intelligence (AI) techniques, with a particular focus on Neural Networks, to enhance the safety of roadside guardrails along the highways of the Turin area. Through the analysis of data collected from sensors and cameras mounted on a moving vehicle, the research aims to automate the detection of guardrails and their classification based on type, compliance with current safety regulations, structural integrity, and the presence of possible damage. The ultimate goal is to develop a monitoring database capable of identifying non-compliant or potentially critical guardrails, thus providing valuable support for preventive maintenance strategies and contributing to improved driver safety, even in the event of an accident.

Contents

List of Figures

Chapter 1

Introduction

Road safety represents a central priority for modern transportation systems, where the effectiveness of roadside infrastructure plays a fundamental role in preventing and mitigating accidents.

Longitudinal safety barriers -commonly known as *guardrails*- are essential elements designed to contain or redirect out-of-control vehicles over the edge of the road, thus reducing the severity of road incidents [1]. Their protective performance, however, strongly depends on their physical condition, proper installation, and compliance with technical regulations. Structural deterioration, accidental damage, inadequate setup, or outdated configurations can significantly compromise their ability to safeguard road users.

Traditional inspection practices rely on **periodic manual surveys** conducted by trained personnel. Although widely adopted, these procedures are time-consuming, costly, and often lack systematic coverage. Their effectiveness is inherently limited by subjective assessments and by the impossibility of ensuring high-frequency monitoring over large territorial areas. Therefore, the major critical issues are:

- the high overall **operating cost** (related to both personnel and equipment for the survey);
- the **risks for personnel** employed in the field, especially during inspection activities on high-traffic roads;
- the **lack of objectivity and repeatability** in evaluations, which depend heavily on the experience and judgment of the operators.

These constraints are particularly relevant in extensive metropolitan road networks -such as the Turin highway system - where thousands of barrier segments are installed under heterogeneous environmental, structural, and traffic conditions.
In recent years, research has increasingly explored **automated techniques** to support or partially replace manual inspections. **Computer-vision-based approaches** and **deep learning models** for guardrail inventory and condition evaluation have demonstrated promising results [2]. Other works have developed methods to assign maintenance priority to safety barriers, combining degradation indicators with spatial information [3]. More recently, advanced computer vision techniques integrated with Global Navigation Satellite System (GNSS) data have been proposed for fully automated damage detection and geolocation of guardrails [4]. Despite these advances, several **open problems** remain unresolved. Current solutions often suffer from:

- the lack of reliable integration between visual detection and geographic information;
- the absence of unified pipelines capable of classifying both the type and the structural condition of guardrails;
- limited scalability in producing georeferenced inventories suitable for long-term monitoring;
- difficulties in the transfer, standardization, and long-term management of digital data, especially when integrating results into Building Information Modeling (BIM) environments for maintenance planning.

Furthermore, most existing studies rely on static imagery or controlled acquisition setups, whereas real-world highway environments introduce challenges such as occlusions, variable lighting conditions, heterogeneous barrier morphologies, and non-standard installations.
This thesis aims to address these open issues by developing an **automated pipeline for the detection, classification, assessment, and georeferencing of guardrails** along the highways of the Turin area, acquired from onboard cameras and GNSS sensors mounted on a moving vehicle. Leveraging convolutional neural networks, the system automatically extracts guardrail regions from video frames, distinguishes among different barrier types, and evaluates their apparent structural condition, with

specific attention to deformation and corrosion. Each detected element is then associated with the corresponding GNSS coordinates, generating a structured geospatial database that can be visualised within a GIS environment and queried according to specific maintenance needs.
The work is structured as follows:

1. **Data collection and preparation**: extraction of frames from videos acquired via monocular and stereo cameras, synchronization with the vehicle's GNSS.

2. **Image segmentation**: implementation of a U-Net neural network for automatic identification of guardrail regions in frames.

3. **Classification and evaluation**: differentiation among guardrail types (e.g., double-wave, triple-wave), and evaluation of structural conditions/damages.

4. **Geospatial Integration**: development of a geospatial monitoring database.

The **ultimate goal** of this work is not merely the automated detection of guardrails, but the construction of a georeferenced monitoring tool that enables municipalities or highway operators to map the condition of their roadside barriers and prioritise interventions based on objective indicators. The methodology developed in this thesis contributes to the broader effort towards predictive and data-driven maintenance in road safety, providing a scalable framework that can be extended to other territorial contexts or integrated with open data sources, such as historical street imagery.

Chapter 2

State of art

2.1 Traditional Inspection Methods

Roadside guardrails (also known as *road safety barriers*) have historically been inspected through routine visual surveys by road maintenance personnel.
In Italy, visual inspections are prescribed by technical guidelines and ministerial decrees. For example, *Decreto Ministeriale* 18/02/1992 n.223 (DM 223/1992) defined road safety barriers as devices intended to contain vehicles leaving the roadway in the safest possible manner [5]. This decree required road authorities (ANAS, highway concessionaires, provinces, and large municipalities) to report biennially on the condition and performance of installed barriers [6].
In practice, however, those reporting requirements were largely unmet – few road operators maintained a precise inventory or condition database of their guardrails, and very little information reached the central Ministry [6]. Thus, traditional methods have relied on periodic on-site visual inspections, wherein trained inspectors assess guardrail segments for damage (e.g. dents, missing bolts, corrosion) and compliance with standards. Figure 2.1 shows typical field activities performed by road maintenance personnel during these visual inspections. ANAS (the Italian road agency) developed standardized inspection forms (see *Allegato A* in the ANAS technical handbook) to guide field personnel in evaluating the state of preservation and efficiency of safety barriers across the network [7]. These inspections focus on visible defects such as deformations, missing elements, excessive deflection, and corrosion, which are recorded to prioritize maintenance.

Figure 2.1: Examples of traditional visual inspections of roadside guardrails performed by field technicians.

Regular inspection is crucial because guardrails directly impact road safety – deficiencies can lead to increased risk of severe accidents. The ANAS handbook emphasizes that systematic control and inspection of safety barriers are vital to ensure their effectiveness and to optimize investment in maintenance [7].
In Italy, inspection procedures have been influenced by European norms as well. The early national guidelines (DM 223/1992 and subsequent updates) were eventually aligned with the European Standard EN 1317, which governs road restraint systems. EN 1317 introduced harmonized terminology and crash-testing criteria for barriers, and Italy formally adopted these through *Decreto Ministeriale* 21/06/2004 n.2367 (DM 2367/2004) [8]. As a result, since 2011 all new or replacement road restraint devices (including guardrails) in Italy must be CE-marked in accordance with EN 1317 [8].
Traditional inspection practices in Europe similarly rely on visual methods. Many countries' road agencies schedule routine surveys where crews visually check barrier continuity, alignment, and damage. Guidelines often require prompt inspection after traffic accidents or extreme events (e.g., collisions, landslides) to identify compromised guardrail sections.
In summary, the state-of-practice has been predominantly manual: inspectors walking or driving along roadways, sometimes using paper or digital checklists to log issues. This approach is labor-intensive and subjective, leading to efforts to improve consistency via standardized forms (as done by ANAS) and to augment human

inspections with technology (discussed later). Nonetheless, visual inspection remains the cornerstone of guardrail maintenance management, forming the baseline against which new automated methods are evaluated. The importance of these inspections is underlined by safety audits and regulations: Italian standards explicitly recognize that regular safety inspections help identify and preempt situations potentially leading to accidents ([7]).

2.2 Artificial Intelligence Applied to Road Safety

In recent years, **Artificial Intelligence** (AI) techniques – particularly computer vision using deep learning – have been increasingly applied to automate the detection and evaluation of road safety features such as guardrails. In road safety inspections, Machine Learning/Deep Learning algorithms can be trained on images of road environments to recognize guardrails, classify their condition, or segment their precise location in the image. This capability addresses the limitations of traditional inspections by providing continuous, objective, and high-speed analysis of visual data. Early applications of **Computer Vision** to guardrails (mid-2000s) did not yet employ modern deep learning, but they laid the groundwork for AI use. The first proposals for automatic guardrail detection, such as those of Broggi et al. (2005), were based on the search for long parallel lines obtained by edge detection, which represent the longitudinal profile of the guardrail. The algorithm extracted edges and analyzed the continuity of lines over time, assuming that a guardrail appeared as a sequence of nearly straight parallel edges aligned with the direction of travel [9, 10] (see Figure 2.2).

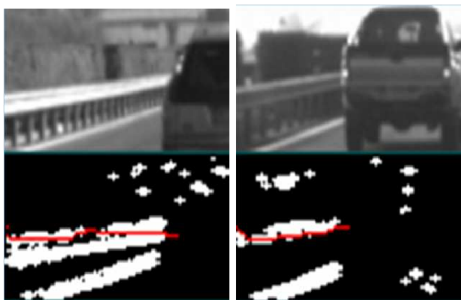


Figure 2.2: Simplified example of guardrail detection in the early work of Broggi et al. (2005), based on the search for parallel lines obtained by edge detection.

Other studies detected guardrails via structure-from-motion (inferring the rail from the camera's egomotion and scene geometry) or template matching[11]. These approaches were often rule-based or relied on hand-crafted features, and they struggled with complex backgrounds, lighting changes, and lack of depth information[12]. As a result, their accuracy was limited in real-world driving scenes[13]. The advent of modern deep learning (around 2012) brought a step-change: convolutional neural networks (CNNs) can automatically learn discriminative features of guardrails (shape, texture, context) from large image datasets, greatly improving detection robustness.
Today, state-of-the-art AI systems for guardrail inspection primarily use deep neural network architectures for **object detection** and **semantic segmentation** tasks.

- **Object detection** locates objects within the image via bounding boxes, assigning each a class (e.g., "guardrail", "vehicle", "pole")- see Figure 2.3. This approach answers the question *"Where is the object?"* providing an approximate delimitation of its position. It is particularly indicated when the goal is to locate or count discrete elements of the scene.
- **Semantic segmentation**, on the other hand, assigns a label to each individual pixel of the image, classifying it into one of the defined categories (e.g., "guardrail", "road", "vegetation", "sky")- see Figure 2.4. The result is a continuous and precise mask, useful for analyzing the shape, length and extension of the guardrail. This approach answers the question *"Which pixels belong to this object?"*.

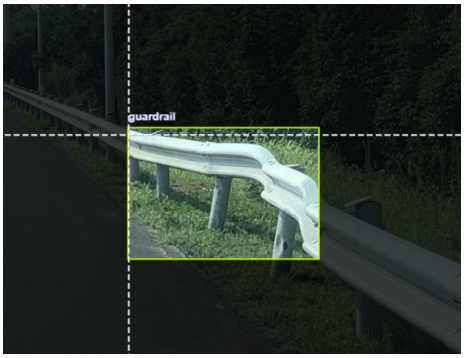


Figure 2.3: Example of *object detection*, where objects (e.g. guardrails) are located via bounding boxes.

Figure 2.4: Example of *semantic segmentation*, where each pixel is assigned to a class (e.g. road, car, sky).

In the case of guardrails, semantic segmentation therefore allows for an accurate delineation of the spatial continuity of the barrier, while object detection is better suited to identifying individual sections or modules of the guardrail present in the image.

Building on this distinction, several deep learning architectures have been developed to address these tasks with high accuracy. Notable examples include *U-Net*, *SegNet*, *DeepLabv3+*, *Faster R-CNN*, and *Mask R-CNN*. Table 2.1 provides a high-level overview of these architectures and the tasks they are typically used for. *U-Net* and *SegNet* are fully convolutional networks designed for semantic segmentation (pixel-wise classification), while *Faster R-CNN* represents a commonly used framework for object detection (predicting bounding boxes and class labels). *Mask R-CNN* extends Faster R-CNN by adding a per-object mask, thus performing instance segmentation. *DeepLabv3+* is another widely used segmentation model, known for its strong performance in complex road scenes and its ability to capture information at multiple spatial scales. All these architectures have been applied to road safety imagery – for example, segmentation networks to outline the continuous length of guardrails in highway images, or Faster R-CNN and Mask R-CNN to detect and mask discrete sections of barrier.

Architecture	**Task**	**Description**
U-Net (2015)	Semantic segmentation	Encoder–decoder network widely used for pixel-wise classification; effective on limited datasets.
SegNet (2015)	Semantic segmentation	Efficient encoder–decoder architecture suitable for real-time or near real-time segmentation.
DeepLabv3+ (2018)	Semantic segmentation	High-performing segmentation model capable of capturing multi-scale information in complex scenes.
Faster R-CNN (2015)	Object detection	Two-stage detector predicting bounding boxes and class labels; strong accuracy in structured environments.
Mask R-CNN (2017)	Instance segmentation	Extension of Faster R-CNN that produces a mask for each detected instance.

Table 2.1: Main deep learning architectures commonly used in road scene analysis.

The most commonly used metrics to evaluate the performance of segmentation and

detection models are the following:

- **Intersection over Union (IoU)**

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}, \tag{2.1}$$

  where A is the predicted mask and B is the ground-truth mask. A value of 1 indicates perfect overlap.

- **Precision and Recall.**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.3}$$

  where TP, FP and FN denote true positives, false positives, and false negatives, respectively. Precision measures whether the model tends to over-segment (many false positives), whereas recall measures its ability to correctly identify all object pixels.

- **F1-score**

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2.4}$$

  which is the harmonic mean of precision and recall. For pixel-wise segmentation, the F1-score is mathematically equivalent to the Dice coefficient.

- **Dice Coefficient**

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}, \tag{2.5}$$

  a commonly used alternative formulation of the F1-score, particularly helpful for segmentation tasks. This coefficient is particularly useful when classes are imbalanced.

- **Pixel-wise Accuracy.**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2.6}$$

where TN denotes true negatives. Although commonly reported, accuracy can be misleading in highly imbalanced segmentation tasks: a model predicting all pixels as background would obtain high accuracy while completely failing at object detection.

- **Cross-Entropy Loss**

$$\mathcal{L}_{CE} = -\sum_{i} y_i \log(\hat{y}_i), \tag{2.7}$$

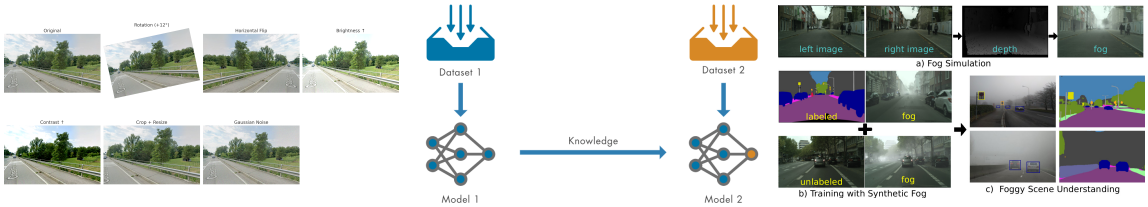
where y_i is the true label and $\hat{y}_i$ the predicted probability for pixel i. Cross-entropy is the most common loss function used during training and drives the model to improve its discrimination between object and background pixels. In the binary case, this reduces to the binary cross-entropy.

2.2.1 Challenges in Training Guardrail Detection Models

Deep learning models require large annotated datasets for training.
A significant challenge in applying AI to guardrail inspection is **dataset scarcity**: there are relatively few publicly available datasets focused on guardrails or barrier damage, especially compared to general road scene datasets. Jin et al. (2024) note that existing guardrail segmentation models in the literature have all been trained on rather limited datasets (often images captured from a vehicle perspective), and there is a lack of diverse training data[14]. This can hinder the models' ability to generalize to different environments (e.g., different countries, backgrounds, or camera viewpoints). For instance, a model trained only on forward-facing highway camera images might perform poorly on side-view or higher-angle images (such as those from drones or mapping vehicles) due to distributional shift. Indeed, studies have observed that certain algorithms which perform well on near-field images lose accuracy on far-field scenes or under different lighting conditions[15]. Overcoming this requires **data augmentation, transfer learning,** or **synthesizing training data** to cover a wide range of scenarios:

- **Data augmentation** consists of artificially increasing the size and variability of the dataset by applying transformations to existing images (e.g., rotations, flips, brightness variations, cropping, noise). This helps the model become more robust to changes in viewpoint and lighting.

- **Transfer learning** uses a model pre-trained on a large, generic dataset and adapts it to the required detection task (e.g., guardrail detection). This allows the model to achieve good performance even with relatively small domain-specific datasets.
- **Synthetic data generation** creates artificial training images, typically through simulation environments or generative models. Synthetic data can represent rare conditions (e.g., unusual lighting, uncommon guardrail types, extreme weather) and viewpoints that would be difficult or expensive to capture in the real world.



(a) Data augmentation (b) Transfer learning (c) Synthetic data

Figure 2.5: Illustration of techniques used to address dataset scarcity: **(a)** example of data augmentation applied to a guardrail image; **(b)** conceptual diagram of transfer learning, where a pre-trained model is adapted to the guardrail detection task; **(c)** synthetic data generation (foggy generation) using simulated images.

Despite the challenges previously cited, recent research results are promising. Using an improved U-Net architecture, Jin et al. achieved a mean IoU above 85% in guardrail segmentation, an 8.63% increase over a standard U-Net, along with higher precision and recall. Their F1-score reached 0.90, about 0.16 higher than the baseline[14, 16]. Such performance indicates that AI models can detect guardrail presence and even segment fine details (like support posts or rail gaps) with high accuracy. Furthermore, AI can not only locate guardrails but also assess their condition. By training on examples of damaged vs. undamaged rails, models have been developed to automatically flag sections with deformities, missing components, or corrosion. Some approaches use a two-step pipeline: first segment the guardrail in the image, then within that region apply classifiers or anomaly detectors to identify specific defects (e.g., a break, or a detached rail)[17, 18]. Performance for damage detection

is commonly reported with precision/recall; for example, above 85% accuracy in identifying certain defect types[18].
Beyond dataset limitations, another key factor influencing performance is the ability of AI models to maintain **robustness** and **generalization**. Road environments are highly variable: guardrails come in different materials and shapes (double-wave, three-wave, concrete barriers, etc), and background scenes can include clutter (vegetation, vehicles, shadows). Models must handle these variations as well as adverse conditions like rain, fog, or nighttime.
To improve generalization, researchers have experimented with **data fusion** (e.g., combining camera data with LiDAR or radar) and domain adaptation. For instance, a vision+radar fusion algorithm was shown to reliably detect guardrails and ignore them as non-threat obstacles in driver assistance systems[19, 20] – the radar provides distance information that confirms a long linear reflector is likely a guardrail. While such sensor fusion targets different objectives (e.g., filtering false positives for vehicle detection), it demonstrates how multiple data sources can reinforce AI's understanding. In the context of automated inspection, combining 2D imagery with 3D depth (from stereo cameras or laser scanners) can significantly improve detection of highway assets like guardrails[21, 22].
In summary, the application of AI to guardrail inspection is a key part of a broader trend of digital road asset management, which aims to continuously monitor infrastructure using imagery and sensor data, with machine learning providing the analytical engine to interpret that data in real time.

2.3 Road Data Acquisition

AI-based guardrail inspection requires robust data **acquisition systems** capable of collecting high-quality images or videos of roadways.
In practice, this is done via specific platforms mounted on a moving vehicle, called **mobile mapping systems (MMS)**. The underlying concept is simple: instead of performing slow and dangerous manual surveys, an MMS integrates **multiple sensors** (cameras, laser scanners, GNSS, etc.) on the vehicle, continuously recording the road environment as it travels along the network. This enables the acquisition of high-resolution images even at road speeds, provides continuous and systematic coverage

of the guardrails, and assigns to each image the precise position and orientation of the vehicle.

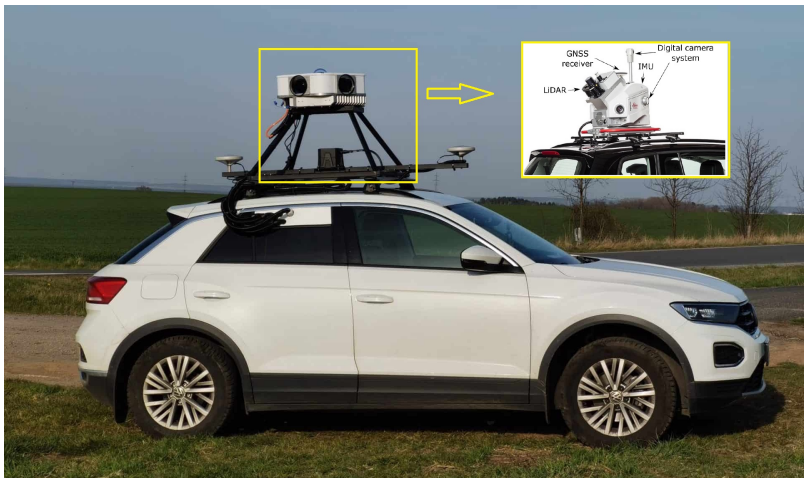


Figure 2.6: Example of a Mobile Mapping System (MMS) equipped with multi-sensor instrumentation.

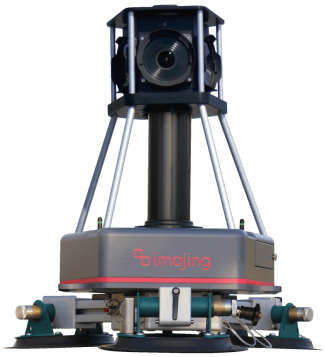
Modern MMS platforms can integrate different sensors depending on the level of detail and accuracy required for road asset inspection. These technologies include:

- **RGB and high-resolution cameras**, which capture detailed visual information on guardrails, posts and roadside elements;
- **Stereo or multi-camera systems**, used to estimate depth, reconstruct portions of the scene in 3D, and improve robustness against lighting variability;
- **LiDAR scanners**, which generate dense 3D point clouds of the road corridor. Guardrails appear as continuous, elevated linear structures, making LiDAR particularly suitable for geometric reconstruction, deformation analysis and structural measurements;
- **360° panoramic cameras**, increasingly used for large-scale road inventories, as they provide complete environmental coverage from a single sensor.
- **GNSS receivers** and **Inertial Measurement Units (IMUs)** to ensure accurate georeferencing of all acquired data by fusing satellite navigation with inertial measurements.

(a) Example of RGB camera

(b) Example of stereo camera

(c) Example of LiDAR scanner



(d) Example of 360° panoramic camera

(e) Example of GNSS receiver

Figure 2.7: Examples of sensing technologies commonly used in modern Mobile Mapping Systems: (a) RGB camera, (b) stereo camera, (c) LiDAR scanner, (d) 360° panoramic camera, (e) GNSS receiver.

Complementary acquisition methods include static or low-speed imaging (e.g., at bridges or hazardous curves) and aerial acquisition using UAVs. Drones can capture high-resolution imagery in areas difficult to access by road vehicles, provided that images are geo-tagged through the drone GPS and ground control points.
Among the most widely used MMS platforms are the **Automatic Road Analyzer (ARAN)** and various professional mapping vans operated by road management authorities.

Figure 2.8: Example of Automatic Road Analyze.

A key element of these platforms is the integration of cameras with **GNSS receivers** and **Inertial Measurement Units (IMUs)**. An IMU includes accelerometers, which measure linear accelerations; gyroscopes, which detect angular velocity, and sometimes magnetometers which provide orientation relative to the Earth's magnetic field. These sensors enable the estimation of the vehicle's roll, pitch, and yaw, the detection of rapid changes in motion, and the reconstruction of the vehicle's trajectory even when GNSS coverage is weak. The fusion of GNSS and IMU measurements ultimately allows each image to be precisely georeferenced, associating accurate spatial coordinates and orientation to every frame. Data acquired via MMS platforms can then be integrated into existing geographic information systems (GIS), enabling large-scale mapping of road safety resources. This integration becomes particularly valuable when combined with institutional spatial databases- such as the Turin Metropolitan City Road Cadastre [23], in our case - which provide authoritative geospatial layers for the regional road network.

Such MMS-based acquisition pipelines are already employed in several large-scale road management projects. A significant example is the mobile mapping campaign conducted by *Autovie Venete* in Italy. A high-performance vehicle, known as MOMAS, was used to survey the entire highway network at speeds of 50–70 km/h [24]. The vehicle continuously recorded GNSS/IMU trajectories and archived georeferenced high-definition images of both sides of the carriageway. These images were later processed through a dedicated Road Management System, which synchronized the visual data with the navigation trajectory, allowing operators to annotate guardrail elements directly within their geographic context [25]. The final output consisted of GIS-ready georeferenced polylines describing guardrail segments.

2.4 Classification of Guardrails in Italy and Regulatory Standards

Guardrails are classified according to their performance and design characteristics in both Italian regulations and European standards. Understanding these classifications is important for evaluating compliance and for mapping the types of guardrails that an automated inspection system may encounter.

2.4.1 Italian Ministerial Decrees

The foundational Italian regulation, DM 223/1992, established an initial classification of road safety barriers based on an **impact severity index**, essentially related to the energy absorption capacity of the barrier. It defined classes A1, A2, A3 for barriers of increasing containment levels for light vehicles, and classes B1, B2, B3 for those designed to contain heavier vehicles [26, 27]. For example, a class A1 barrier was one allowing a severity index up to 15 kJ, A2 up to 50 kJ, A3 up to 150 kJ; whereas B1 allowed up to 300 kJ, B2 up to 600 kJ, and B3 up to 1000 kJ [26, 27]. In practical terms, these classes corresponded to increasing strength: A-class barriers (often single or double steel beam rails) for cars and small vehicles, B-class (stronger) for buses, trucks, etc. The decree also tied minimum barrier classes to road types – for instance, certain extra-urban highways required at least class A3 or B1 depending on traffic and road category [28]. This schema was unique to Italy at the time, preceding the harmonized European system. To clarify the differences between the former Italian classification and the current European framework, Table 2.2 summarizes the main containment classes defined by DM 223/1992 and the corresponding EN 1317 levels.

Regulation	Class/Level	Energy/Test Reference	Test Vehicle
DM 223/1992	A1	$\leq$ 15 kJ	Passenger car
	A2	$\leq$ 50 kJ	Passenger car
	A3	$\leq$ 150 kJ	Passenger car
	B1	$\leq$ 300 kJ	Heavy vehicles
	B2	$\leq$ 600 kJ	Heavy vehicles
	B3	$\leq$ 1000 kJ	Heavy vehicles
EN 1317	N1, N2	Crash tests with cars (up to 1500 kg)	Passenger car
	H1, H2, H3, H4	Tests with heavy vehicles and cars at higher speeds	Truck, bus
	L1, L2	Low containment levels	Light passenger car
	(Special parapets)	Dedicated test procedures	Various

Table 2.2: Comparison between the DM 223/1992 classes and the EN 1317 containment levels

DM 2367/2004 updated the Italian classification by adopting the EN 1317 standard. Article 1 of DM 2367/2004 explicitly replaces the old technical instructions with new ones conforming to EN 1317 parts 1–4 [29]. EN 1317 introduced standardized containment levels such as N1, N2 (*Normal containment*) and H1, H2, H3, H4 (*Higher containment*), as well as L1, L2, etc. (*Low containment*) and special levels for bridge parapets. Each level corresponds to a crash test involving specific vehicle types, masses, and speeds. For example, an H2 barrier is tested to contain a 13-ton bus at 70 km/h and a car at 110 km/h [30].

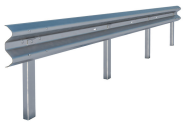
The adoption of EN 1317 meant that Italy's categorization shifted to these codes; Italian road projects and inspections now refer to barriers as "H2", "H3", etc., rather than A/B classes. The ANAS Technical Manual notes that the European framework comprises 8 parts of EN 1317, covering terminology, impact test procedures, performance classes for barriers, crash cushions, terminals, transitions, product certification, etc. [31]. A key aspect introduced by EN 1317 is the requirement for crash testing and CE certification – since 2011, only barriers that have been crash-tested according to EN 1317 and have CE marking (issued by a notified body) can be installed on Italian and European roads [32]. This ensures a uniform level of safety performance. Notably, DM 2367/2004 (Art. 6) mandated that *attenuatori d'urto* (crash cushions) be used to shield fixed obstacles on roads, as per EN 1317-3, and that all new barriers meet the updated criteria [29].

2.4.2 Guardrail Typologies

Beyond performance classes, guardrails can be categorized by their **structural type** and **material**.

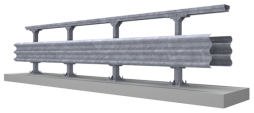
The two primary families are **rigid** systems and **deformable** systems. Rigid barriers are typically made of concrete (e.g. the New Jersey or concrete step barrier); they essentially do not deflect under impact but dissipate energy through the vehicle's deformation and friction. Semi-rigid barriers are the common steel guardrails – usually galvanized steel beams (double-wave or triple-wave profiles) mounted on posts – which deflect upon impact to absorb kinetic energy. Flexible barriers refer to high-tension wire rope barriers (cable barriers) which deflect the most during a crash [33].

Italian practice historically centered on steel guardrails as the standard solution along highways and rural roads, with concrete barriers used in medians of high-speed roads or where space is limited. Cable barriers have seen more limited use in Italy but are employed in other European countries. Figure 2.9 illustrates the barrier types most commonly installed on Italian highways, including double-wave and triple-wave steel guardrails, concrete median barriers, and bridge parapets.



(a) Double-wave

(b) Triple-wave



(c) Concrete median

(d) Parapet

Figure 2.9: Example of most common guardrail types used on Italian motorways: (a) double-wave steel barrier, (b) triple-wave barrier, (c) concrete central median barrier and (d) bridge parapet barrier.

Category	**Material**	**Distinctive Features**	**Typical EN 1317 Class**
Double-wave steel guardrail	Galvanized steel	Two horizontal corrugations; used as roadside or median barrier; often equipped with retroreflectors	N2 – H2
Triple-wave steel guardrail	Galvanized steel	Three corrugations; higher stiffness; may include an upper sound-absorbing panel	H1 – H2
Concrete "New Jersey" barrier	Precast concrete (HDPE for temporary works)	Modular monolithic profile; used as permanent median barrier or for temporary delineation in work zones	N2 (permanent), T1–T3 (temporary)
Laminated timber / corten barrier	Timber with steel core (or corten steel)	Aesthetic solution used in scenic or regional roads; combines wooden appearance with metal structural strength	H1
Central median steel barrier	Galvanized steel	Double-faced configuration based on double-wave beams mounted on central posts	H2
Bridge parapet barrier (PS1/PS2)	Steel or reinforced concrete	Anchored to bridge decks; requires extended anchorage beyond the structure edges	H2 – H3
Temporary modular barrier	HDPE or lightweight plastic	Lightweight interlocking modules used in construction zones for temporary traffic channelization	T1 – T3

Table 2.3: Main guardrail types used in Italy and their typical properties

In addition to the visual examples shown above, Table 2.3 summarizes the main guardrail categories used on Italian highways, detailing their materials, structural characteristics, and typical EN 1317 containment classes.
In summary, Italy's guardrails are categorized by containment performance (old A/B system; now EN 1317 N/H classes) and by structural type (rigid, semi-rigid, flexible). Automated AI-based inspection can help identify mismatches between required and installed classes, and detect defects that may reduce performance [34].

Chapter 3

Methodology

3.1 Innovative Contributions

The proposed work introduces innovative contributions to currently available approaches for road guardrail inspection. The key aspects are summarised below:

- **Guardrail-focused detection pipeline:** Most existing studies treat guardrails as secondary objects within more general detection frameworks (e.g., road elements detection, vehicle detection). In contrast, this thesis proposes a pipeline completely dedicated to guardrail detection, enabling significantly more accurate segmentation of this specific class.
- **Low-cost data acquisition system:** Many commercial MMS (such as Leica Pegasus, Trimble MX, or Topcon IP-S) are effective but extremely expensive and proprietary. The solution proposed in this work relies instead on low-cost, low-power sensors mounted on a standard vehicle.
- **Spatio-temporal synchronisation:** Several studies perform image detection without integrating the results into the geographical domain. In contrast, this thesis implements a workflow for synchronising video frames with navigation data.
- **Creation of a geo-referenced guardrail inventory:** Public administrations currently lack an open-source database containing information on guardrail

location, compliance, and typology. The final product of this work, therefore, represents a valuable tool for maintenance planning and safety assessment.

- **Integration of Google Street View:** An innovative aspect of this work is the use of Google Street View as an additional open-data source to validate, compare, and monitor the condition of guardrails over time.
- **Modular and replicable framework:** Unlike proprietary systems, the entire pipeline is designed to be modular and easily replicable by public authorities or research institutions.

3.2 Proposed approach

The primary objective of this thesis is to develop an automated, robust, and scalable methodology for the comprehensive **identification, classification, and condition assessment** of roadside barriers **(guardrails)**. This approach is specifically tailored to address the limitations of conventional manual inspection methods, offering an objective and continuous evaluation of road safety infrastructure. The case study focuses on a specific highway segment near Turin, Italy.
The proposed approach integrates state-of-the-art techniques from **Artificial Intelligence (AI)**, advanced **Computer Vision**, and **Georeferencing** within a structured and modular pipeline. This architecture is explicitly designed to operate effectively under common real-world driving scenarios.
The methodology is organized into a **sequential flow of processing stages** that systematically transform raw sensor data into meaningful and spatially localized information for road safety assessment. Starting from the acquisition of video and GNSS data, the pipeline encompasses **five distinct macro-phases**, each subdivided into specific tasks.

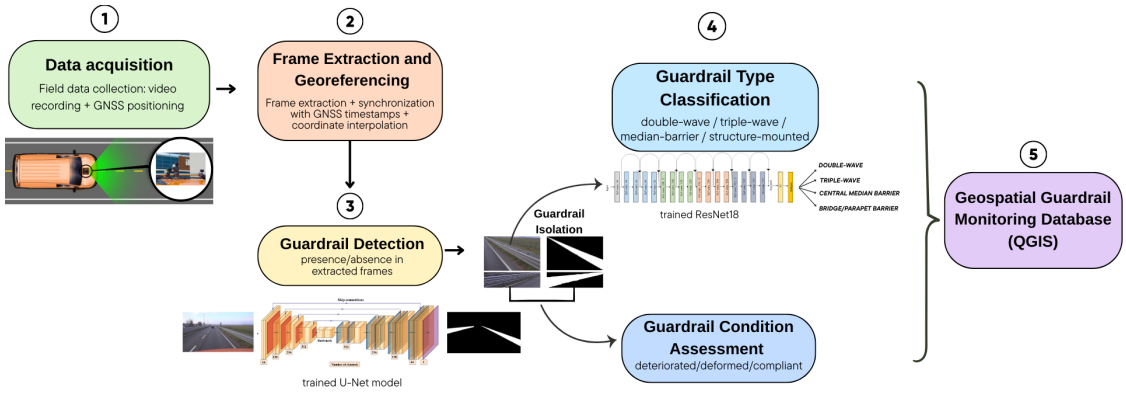


Figure 3.1: Schematic representation of the methodology.

These macro-phases are:

1. **Data Acquisition**
2. **Frame Extraction and Georeferencing**
3. **Guardrail Detection**
4. **Guardrail Type Classification and Condition Assessment**
5. **Geospatial Guardrail Monitoring Database Creation**

At a high level, the proposed pipeline operates as follows. Starting from raw video data acquired through onboard cameras and synchronized GNSS tracks, representative frames are extracted and partially annotated to generate binary masks of roadside barriers. These masks, together with external datasets, are used to train a neural network for guardrail segmentation, enabling the automatic detection and localization of barrier elements within each frame. A second neural network is then employed to classify each identified guardrail segment by type, while additional image-based analyses support the assessment of potential damage, such as deformities or surface deterioration. Finally, all extracted attributes—combined with geospatial coordinates obtained during the georeferencing stage—are integrated into a structured geospatial database suitable for visualization and analysis within GIS environments such as QGIS.

Figure 3.1 provides a **schematic representation of the entire workflow**, serving as a conceptual guide for the detailed descriptions presented in the subsequent sections of this chapter. Each module addresses a specific task, contributing to the overall goal of generating an interpretable and spatially coherent representation of the guardrail

conditions along the analyzed road segment. This modularity ensures **flexibility, maintainability,** and **scalability** for future development or adaptation.

3.3 Data acquisition

The first phase of the methodology involves the **collection of high-resolution visual and geospatial data** from the roadside environment. Figure 3.2 shows this first block of the pipeline. At the end of this step, raw video footage and positioning data, provided by a receiver, are available for the subsequent processing stages.

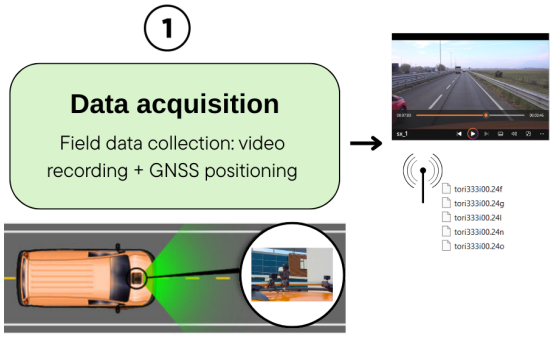


Figure 3.2: Data acquisition block, where video streams and GNSS trajectories are collected from the mobile sensing platform.

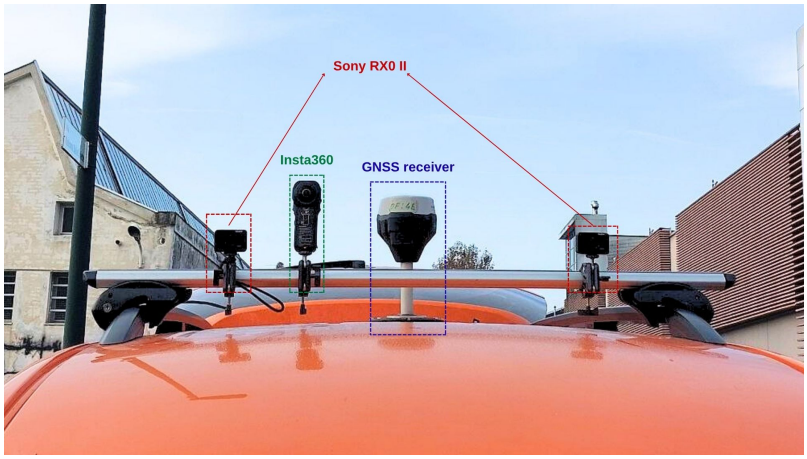


Figure 3.3: Multi-sensor data acquisition setup, including stereo Sony cameras, Insta360 panoramic camera, and a GNSS receiver.

A purpose-built **multi-sensor data acquisition platform**, rigidly mounted on the roof of a dedicated survey vehicle. The system integrates complementary sensors—stereo RGB cameras, a panoramic camera, and a GNSS receiver—operating in parallel and synchronized at the system level. This setup enables the construction of a rich and temporally aligned dataset suitable for the subsequent tasks.
As shown in Figure 3.3, the platform consists of:

- **Two stereo RGB cameras (Sony RX0 II)** mounted rigidly on a custom aluminium crossbar with a fixed baseline of approximately 25-30 cm. Each camera captures video at up to 25 fps with a wide dynamic range.
- **A 360-degree panoramic camera (Insta360 One X2)**, positioned centrally on the crossbar to ensure uniform coverage.
- **A GNSS receiver**, installed close to the geometric center of the crossbar to minimize offset from the optical sensors. The receiver supports dual-frequency, multi-constellation GNSS and records Receiver Independent Exchange Format (RINEX) raw measurements. This enables centimeter-level accuracy in trajectory reconstruction and precise georeferencing of each video frame.

All devices were synchronized using system-level timestamps to ensure consistent temporal alignment across data streams.
Table 3.1 summarizes the main characteristics, strengths, and limitations of the selected sensors, clarifying the rationale behind the acquisition setup.

Sensor	**Key Specs**	**Res./FPS**	**Strengths**	**Cost (EUR)**
Sony RX0 II (×2)	1/2.3" CMOS; 24 mm eq.; stereo baseline 25–30 cm	4K@25fps	High dynamic range; compact; robust; stereo depth	750–800 each
Insta360 One X2	360° FOV; dual fisheye lenses	5.7K@25fps	Full environmental coverage; useful context	450–500
ZED-F9P GNSS	Dual-frequency L1/L2; multi-constellation	20 Hz raw	Centimeter-level accuracy (RTK/PPK)	200–300

Table 3.1: Summary of the sensors used in the acquisition platform.

3.4 Video Frames Extraction and Georeferencing

Figure3.4 illustrates the second block of the pipeline. In this phase, the recorded video and the GNSS data collected by the receiver are processed to extract the individual frames. Each frame is then associated with a corresponding geographic position through timestamp synchronization and coordinate interpolation. The output of this step is a set of georeferenced frames that serve as input for the subsequent analysis tasks.

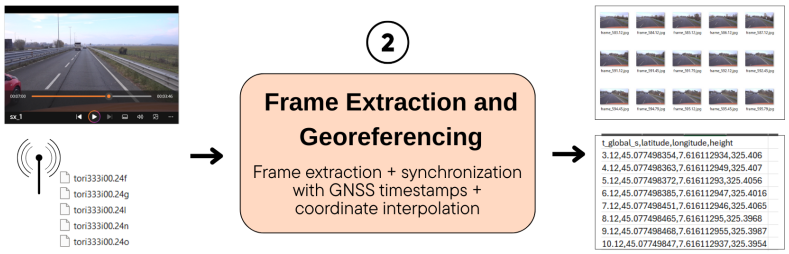


Figure 3.4: Second block of the pipeline: extraction of video frames and their georeferencing.

In this phase, a **multi-step procedure** is adopted to extract meaningful video frames and accurately assign geographic coordinates to each of them. First, the RINEX file is processed to obtain a refined GNSS trajectory. Then, the vehicle's speed profile is analyzed to design an optimized frame-extraction strategy, avoiding redundancy and ensuring that frames are uniformly distributed along the road—rather than simply sampled at a constant temporal rate. Finally, a time-based georeferencing procedure is implemented: video timestamps are synchronized with GNSS timestamps, and each extracted frame is assigned the corresponding geographic position based on temporal alignment.

3.4.1 GNSS Data Processing

The **raw GNSS data were processed** using the open-source software `RTKLib`, applying a *post-processing kinematic* (PPK) approach. Differential correction enhances GNSS positioning accuracy by exploiting the fact that two receivers observing the same satellites at nearly the same time share most of the atmospheric and satellite-

originated errors. A reference *base station* with known coordinates is used to compare its measurements with those of the *rover* mounted on the vehicle.
For each satellite, both receivers collect pseudorange and carrier-phase observations, which contain the true geometric distance plus several error sources (ionospheric and tropospheric delays, satellite clock drift, multipath, and receiver noise). Because the base position is known, its measurement errors can be estimated. By forming **single differences** between rover and base observations, most common-mode errors cancel out; computing **double differences** between satellites further removes satellite clock errors and minimizes residual atmospheric effects.
This differential correction technique improves positional accuracy by minimizing common-mode errors between the rover (vehicle) and a reference base station, achieving **centimeter-level precision** in stable segments. The output of this processing is a `.pos` (see Figure 3.5) file containing, for each epoch, the estimated geographic coordinates (latitude, longitude, altitude), the corresponding timestamp, a fix-quality indicator (Q), and several accuracy metrics such as standard deviations and the number of visible satellites.
These data served as the basis for both speed computation and time synchronization with video frames, ensuring spatial and temporal alignment between the vehicle's motion and the recorded imagery.

```
                          PSEUDORANGE CONSTRUCTION
% program   : RTKPOST ver.2.4.3 b34
% inp file  : C:\Users\attam\Documents\Tesi\dati GNSS
rilievo\reach_raw_202411280914.24O
% inp file  : C:\Users\attam\Documents\Tesi\dati GNSS
rilievo\tori333i00.rnx\tori333i00.24o
% inp file  : C:\Users\attam\Documents\Tesi\dati GNSS
rilievo\tori333i00.rnx\tori333i00.24n
% inp file  : C:\Users\attam\Documents\Tesi\dati GNSS
rilievo\tori333i00.rnx\tori333i00.24g
% obs start : 2024/11/28 09:15:15.0 GPST (week2342 378915.0s)
% obs end   : 2024/11/28 09:42:52.0 GPST (week2342 380572.0s)
% pos mode  : Kinematic
% freqs     : L1+2
% solution  : Combined
% elev mask : 15.0 deg
% dynamics  : off
% tidecorr  : off
% ionos opt : Broadcast
% tropo opt : Saastamoinen
% ephemeris : Broadcast
% navi sys  : GPS GLONASS
% amb res   : Fix and Hold
% amb glo   : OFF
% val thres : 3.0
% antenna1  :                      ( 0.0000  0.0000  1.5800)
% antenna2  :                      ( 0.0000  0.0000  0.0000)
% ref pos   : 45.063364647    7.661277676   310.8200
%
%
(lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,n
s=# of satellites)
%  GPST                  latitude(deg) longitude(deg)  height(m)   Q  ns
sdn(m)   sde(m)   sdu(m)  sdne(m)  sdeu(m)  sdun(m) age(s)  ratio
2024/11/28 09:15:15.000   45.077498406    7.616112925   325.4161   1  14
0.0044   0.0041   0.0104   0.0015  -0.0022  -0.0038   0.00  386.7
```

Figure 3.5: Excerpt of the RTKLib `.pos` output file, showing the processing configuration and the structure of the corrected PPK solution.

3.4.2 Speed profile

Based on the `.pos` file, the vehicle's speed profile was computed as follows. For each consecutive pair of GNSS points, the three-dimensional distance was obtained by combining the horizontal geodesic distance with the vertical elevation difference:

$$d^{(i)} = \sqrt{\left(d_{2D}^{(i)}\right)^2 + (h_{i+1} - h_i)^2} \tag{3.1}$$

where $d_{2D}^{(i)}$ is the geodesic distance between two points, and h_i, h_{i+1} are their respective altitudes. The geodesic distance was computed using the `geopy` library, which implements the Karney (2013) algorithm on the WGS-84 ellipsoidal Earth model, ensuring high precision and convergence. The instantaneous speed was then derived as:

$$v^{(i)} = \frac{d^{(i)}}{t_{i+1} - t_i} \tag{3.2}$$

with $v^{(i)}$ expressed in meters per second and later converted to kilometers per hour. The resulting speed profile was subsequently used to design an adaptive frame extraction strategy: instead of sampling frames at a constant rate (e.g., 1 frame per second), a variable-rate approach was adopted. In segments where the vehicle moved faster, more frames were extracted to maintain adequate spatial resolution; conversely, in slower or stationary segments, fewer frames were sampled to reduce redundancy.

3.4.3 Video–GNSS Synchronization

To accurately georeference each extracted frame, it was necessary to synchronize the video and GNSS timelines. The synchronization point was identified using a video frame in which a smartphone displaying the current time appeared. The timestamp visible on the phone served as a temporal reference and was matched to the corresponding timestamp in the RINEX-derived `.pos` file. Once the frame rate (*frames per second*, FPS) of the video was verified (e.g., 25 fps), each frame's timestamp could be calculated according to:

$$t_i = t_{ref} + \frac{(i - N)}{FPS} \tag{3.3}$$

where t_{ref} is the known timestamp of the reference frame N. This allowed each frame to be associated with a precise acquisition time expressed in GNSS time.

3.4.4 Spatial Interpolation of Frame Coordinates

Since video frames typically correspond to timestamps not explicitly present in the GNSS log (which usually records positions at 1 Hz), an interpolation method was applied to estimate the geographic coordinates of intermediate frames. For each GNSS time interval $[t_i,\, t_{i+1})$, where $\mathbf{P}_i = [\phi_i, \lambda_i, h_i]$ denotes latitude, longitude, and altitude, the total displacement $\Delta\mathbf{P} = \mathbf{P}_{i+1} - \mathbf{P}_i$ was divided according to the number of frames M to be extracted within that interval.
Coordinates were assigned by evaluating fractional increments of the displacement vector, excluding the final boundary point to avoid duplication between consecutive intervals. The position of the k-th frame within $[t_i,\, t_{i+1})$ was therefore computed as:

$$\mathbf{P}(t_{i,k}) = \mathbf{P}_i + \frac{k}{M}\,\Delta\mathbf{P}, \qquad k = 0, 1, \ldots, M-1. \tag{3.4}$$

This formulation ensures that the last frame in the interval approaches, but does not overlap with, the next GNSS epoch. The interpolation was applied independently to latitude, longitude, and altitude, resulting in spatially consistent coordinates for all extracted frames.

3.5 Guardrail Detection

The third stage of the proposed pipeline, Guardrail Detection, represents a critical requirement of this thesis, due to the inherent challenges in accurately identifying road safety barriers within video frames. To address this, we adopted an image segmentation approach using deep learning, specifically employing a **U-Net architecture**. U-Net is selected for its proven efficacy in pixel-level segmentation, particularly in scenarios constrained by limited dataset sizes.
Figure3.6 illustrates the process underlying the guardrail detection step. After extracting the individual frames from the input videos (from the previous stage of the pipeline), a subset of these frames — enriched with images from an external dataset to ensure a wider variety of environmental conditions — was used to train the neural

network. For all selected frames, a binary mask was previously generated to serve as the ground truth input during the training phase.

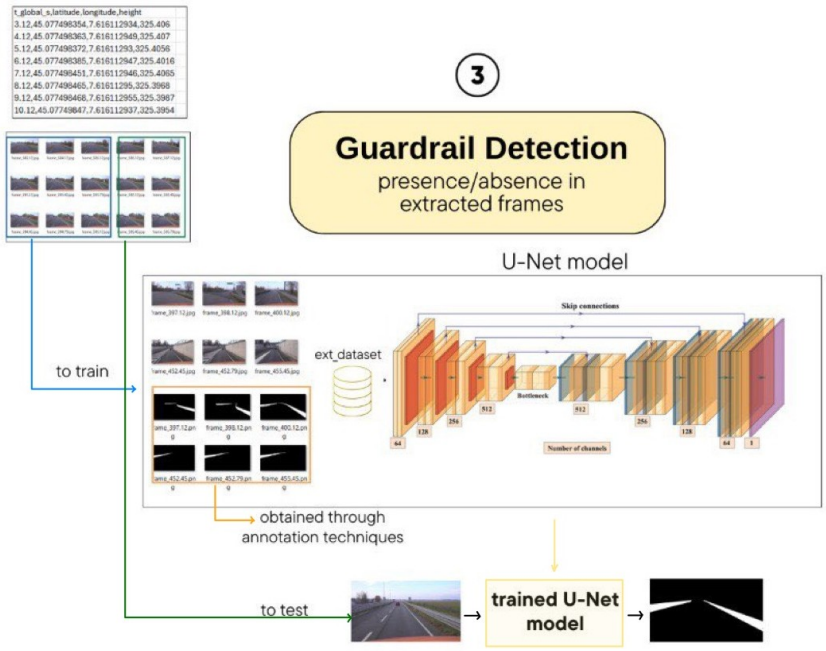


Figure 3.6: Third block of the pipeline: guardrail detection.

Once trained, the neural network is applied to the remaining frames, for which no mask is available. The goal is to enable the model to automatically detect the presence of guardrails in the scene, producing for each image a **binary segmentation mask** in which the pixels belonging to the guardrail are clearly distinguished from the background.

3.5.1 Dataset Generation and Annotation Strategy

A significant hurdle in this phase was the absence of ground truth data (i.e., pre-existing binary masks corresponding to the video frames). To address this, we initially explored two strategies: manual labeling and a semi-automatic pipeline based on classical computer vision.

- **Manual Annotation:** utilizing the MATLAB *Image Labeler* tool, we manually delineated guardrail regions on a representative subset of frames. While this

method guarantees high-precision ground truth, it is computationally expensive and labor-intensive due to the requirement for pixel-perfect tracing.

- **Semi-Automatic Generation via Classical Computer Vision:** A prototype pipeline was developed based on **Canny edge detection** and the **Hough Transform** to detect the linear structures typical of guardrails. A **Kalman filter** was then employed to track these features across frames to reduce temporal noise. Despite its speed, this method required substantial parameter tuning (e.g., adaptive thresholding for varying illumination) and often produced masks with insufficient accuracy for training deep learning models.

To better illustrate the behaviour of the **semi-automatic pipeline**, Figure 3.7 reports a sequence of representative frames processed through all intermediate steps. Each column corresponds to a different stage of the workflow, allowing a clear understanding of how geometric constraints and color filtering jointly contribute to the extraction of preliminary guardrail masks.

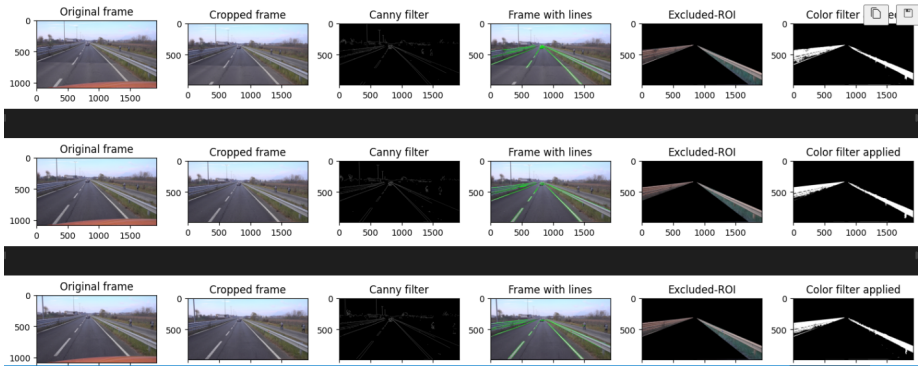


Figure 3.7: Example of semi-automatic segmentation applied on a sequence of consecutive frames.

The main stages are summarised below.

1. **Frame Cropping.** The upper portion of each frame is removed to discard irrelevant background (sky, distant landscape), which typically introduces clutter in edge and line detection. This step focuses the analysis on the roadway and adjacent infrastructure.

2. **Edge Detection.** A Canny operator is applied to extract strong structural gradients. Guardrails generally produce long, sharp edges due to their metallic profile and perspective projection. Edge detection, therefore, provides the first reliable cue for identifying regions with potential guardrail presence.

3. **Line Detection with Hough Transform.** The edge map is processed with the (probabilistic) Hough Transform to extract straight-line segments. A filtering strategy based on line orientation and image position removes spurious detections (e.g., lane markings, vertical poles) while preserving the slanted lines characteristic of guardrails. This step produces a set of geometric primitives that encode the scene's perspective structure.

4. **Vanishing-Point Estimation.** Selected lines are combined to estimate a vanishing point using a RANSAC-based procedure. The vanishing point captures the global image geometry and provides a constraint for understanding which regions of the frame are compatible with guardrail placement. This information is used to define admissible regions of interest (ROIs).

5. **Geometric ROI Construction.** Using the vanishing point and the extremal detected lines, polygonal regions where guardrails cannot be present are removed (e.g., the road centre, far-field upper areas, or internal roadside zones). This significantly reduces false positives and restricts the subsequent processing to the most relevant image areas.

6. **Color-Based Filtering.** The remaining portion of the frame is converted to the HSV color space, and thresholding is applied to isolate metallic grey tones commonly associated with steel guardrails.

7. **Temporal Stabilisation.** A Kalman filter is employed to stabilise line positions and the vanishing point over time, reducing jitter between consecutive frames and improving the temporal consistency of the generated masks.

The final mask is obtained by combining geometric constraints and color filtering. While not accurate enough to serve as definitive ground truth, this mask provides a useful first approximation, later refined manually or replaced by deep-learning-based segmentation.

Limitations of Classical Approaches

While the semi-automatic pipeline provided a rapid way to generate initial masks, it proved insufficient as a standalone solution for the final detection system. Classical edge detection relies heavily on strong gradients and linear approximations. Consequently, it struggled significantly in scenarios involving:

- **Curved Guardrails:** The Hough Transform is optimized for straight lines and often fails to capture the curvature of barriers on winding roads.
- **Complex Backgrounds:** High-frequency textures (e.g., trees, fences) often trigger false positives.
- **Varying Illumination:** Fixed thresholds for edge detection do not generalize well across different lighting conditions (e.g., shadows vs. direct sunlight).

These limitations necessitated the use of a Deep Learning approach capable of learning robust, non-linear semantic features.

3.5.2 Neural Network Architecture: U-Net

To address the binary segmentation task, we selected the **U-Net architecture**. It is particularly effective for tasks with limited training data due to its ability to combine low-level detail with high-level context.
The architecture is characterized by a symmetric encoder-decoder structure:

1. **Encoder (Contracting Path):** Extracts hierarchical features by progressively reducing spatial resolution while increasing feature depth. It captures the "context" of the image (i.e., "what" is present).
2. **Decoder (Expanding Path):** Reconstructs the spatial resolution to generate a precise segmentation mask. It addresses the "localization" (i.e., "where" it is).
3. **Skip Connections:** A critical feature of U-Net is the direct connection between encoder and decoder layers. These connections propagate high-resolution spatial information lost during pooling, ensuring sharp boundaries in the final segmentation.

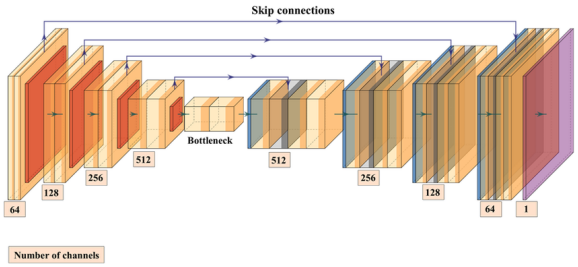


Figure 3.8: Traditional U-Net model architecture.

After training, the model was able to predict accurate guardrail segmentation masks for new frames.

3.5.3 Performance Evaluation Metrics

In road scene imagery, guardrail pixels typically constitute a very small fraction of the total image area (often $< 5\%$). To prevent the model from converging to a trivial solution (i.e., predicting everything as background), we utilized **Binary Focal Loss**. Unlike standard Cross-Entropy, Focal Loss applies a modulating factor to down-weight easy examples (background) and focus training on hard, misclassified examples (guardrails).
The **Binary Focal Loss** is defined as:

$$\mathcal{L}_{\text{focal}} = -\alpha_t(1 - p_t)^\gamma \log(p_t), \tag{3.5}$$

where α_t balances the importance of positive and negative samples, γ controls the focusing strength, and p_t is the predicted probability for the true class. Model performance is evaluated using the metrics described in Section 2.2.

3.6 Guardrail Type Classification

The fourth macro-phase of the proposed methodology involves a parallel workflow: on one side, the classification of guardrails by typology, and on the other, the assessment of defects such as corrosion or structural deformations. Figure 3.9 illustrates the **classification** branch, which is described in detail in the following section.

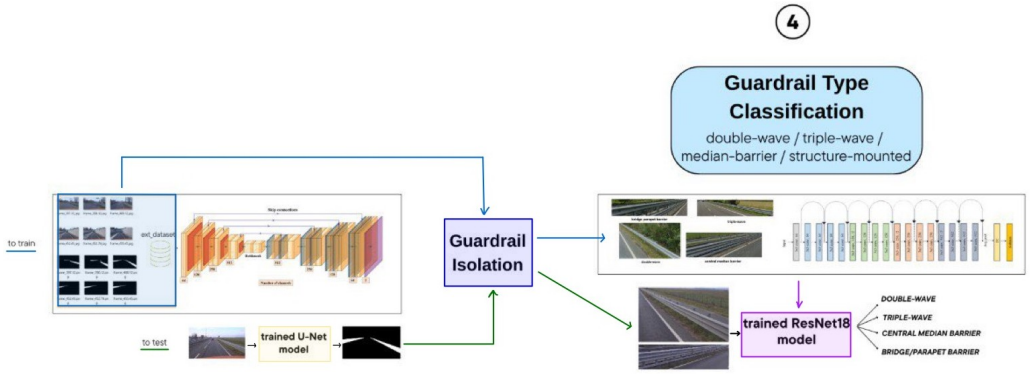


Figure 3.9: Fourth block of the pipeline: guardrail type classification branch.

After the segmentation phase, each detected guardrail must be classified according to the types defined by European and Italian road safety legislation (e.g., EN 1317), discussed in the previous chapter 2.4.2. In this work, 4 main categories were considered:

- **Double-wave guardrail**
- **Triple-wave guardrail**
- **Central median barrier**
- **Bridge/parapet barrier (for engineering structures)**

These categories represent the most commonly used road restraint systems on Italian roads, and their classification is based on both structural characteristics (e.g., wave profile) and functional position (e.g., roadside, central divider, or bridge).
The classification of each guardrail is performed using a convolutional neural network based on the **ResNet-18 architecture**. Before training the model, however, a preliminary guardrail isolation step is required, as shown in Figure 3.9.
Guardrail Isolation. Starting from the original image and its corresponding segmentation mask, each guardrail segment present in the frame is extracted independently. The masks allow for the automatic cropping of the image regions containing the guardrail; when multiple barriers appear in the same frame (for example, one on each side of the roadway), each segment is treated as a separate image. This process produces a dataset of patches focused exclusively on the guardrail area, thereby reducing background noise and optimising the information content for typology classification. This operation is also beneficial for the subsequent stages, as the assessment

of structural conditions is performed directly on the isolated guardrails rather than on the entire frame. For the training phase, the classification network is trained with isolated guardrails extracted from the training images of the previous U-Net segmentation model. Once trained, the classifier is tested on the isolated guardrails obtained from the remaining frames, using the masks generated by the segmentation network in the testing stage.

Guardrail Typology Decision. Each isolated patch is then analysed by the classification model:

- Input: cropped images containing the guardrail;
- Model: a ResNet-18 convolutional neural network, properly trained;
- Output: the typological label assigned to the guardrail segment.

3.6.1 ResNet-18 Architecture and Residual Learning

ResNet-18 is a Convolutional Neural Network (CNN) renowned for its efficiency and performance in image recognition tasks. Figure 3.10 shows the traditional structure of ResNet-18.

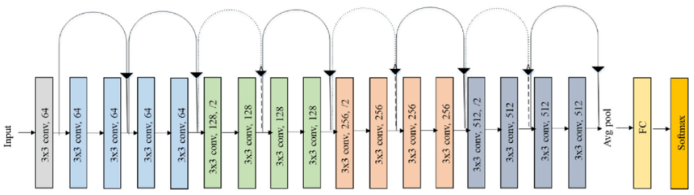


Figure 3.10: Original ResNet-18 Architecture.

This specific variant provides an optimal trade-off between network depth and computational complexity, making it suitable for processing large-scale road inventory datasets. The core innovation of the Residual Network (ResNet) family is the introduction of *skip connections* (or identity shortcuts). In traditional deep networks, increasing depth often leads to the vanishing gradient problem, causing performance degradation.

ResNet addresses this by allowing gradients to flow directly through identity mappings. Formally, rather than learning a direct underlying mapping $H(x)$, the network fits a residual mapping $F(x) = H(x) - x$. The original mapping is then recast into $F(x) + x$.

This architecture enables the training of significantly deeper networks while preserving feature transmission, allowing the model to effectively learn complex hierarchical features—such as profile structures, wave shapes, and panel configurations—that are critical for distinguishing guardrail types.

Transfer Learning and Adaptation

Given the specific nature of guardrail classification and the potential constraints in dataset size compared to massive public datasets, we adopted a **transfer learning** approach. We utilized a ResNet-18 model pre-trained on the ImageNet dataset, leveraging its learned feature extractors (edges, textures, and shapes). ResNet-18 offers an excellent compromise between computational cost and generalization capability, making it well-suited for a moderately sized dataset such as the one obtained in this case study.
To adapt the network for our specific multi-class problem, the final fully-connected (FC) layer was replaced.The original fully connected layer was replaced by a new mapping:

$$\text{fc} : \mathbb{R}^{512} \rightarrow \mathbb{R}^{C}$$

where $C = 4$ corresponds to the number of guardrail typologies considered. All remaining components of the network—including residual blocks and skip connections—were left unchanged.

Data Augmentation Strategy

To enhance model robustness and prevent overfitting, a data augmentation pipeline was integrated into the training phase. Real-world infrastructure imagery is subject to high variability in lighting, weather, and occlusion. Therefore, the network was trained using **transformations** such as:

- **Random horizontal flips** (to simulate bidirectional viewpoints);
- **Photometric distortions** (brightness, contrast, and saturation adjustments);
- **Random rotations**.

These augmentations force the model to focus on invariant structural features rather than incidental pixel-level correlations, ensuring reliable performance even under challenging conditions such as shadows or partial occlusions.

3.7 Guardrail Condition Assessment

After the typological classification of the guardrails, the pipeline proceeds with a parallel branch dedicated to the evaluation of the structural condition of the barriers. Figure 3.11 schematically illustrates this module.

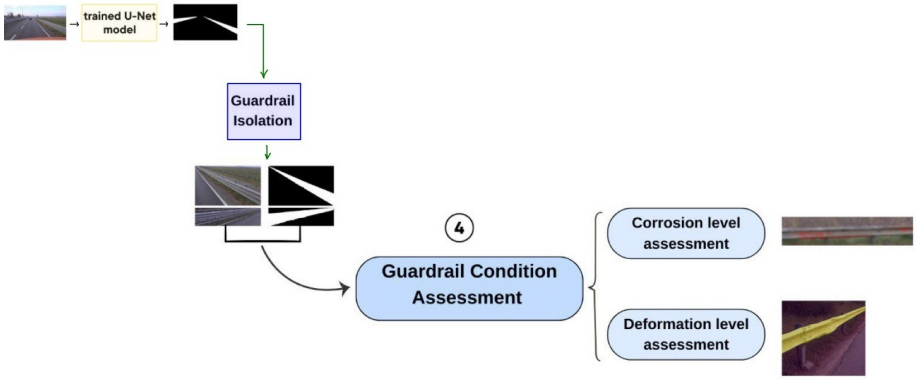


Figure 3.11: Fourth block of the pipeline: guardrail condition assessment branch.

In this stage, the system receives as input the isolated guardrails (together with their corresponding segmentation masks) extracted from the images processed during the testing phase of the U-Net model. The objective is to analyse each segment independently, identifying any anomalies that may compromise the safety performance of the barrier. This block consists of two main sub-modules:

- **Corrosion Assessment** – determines the presence or absence of corrosive phenomena, with particular attention to rust formation on metallic surfaces.
- **Deformation Assessment** – evaluates potential geometric deformations, such as bending, impacts, or torsional distortions, which indicate structural damage to the guardrail.

These two micro-blocks allow the system to integrate, in addition to the guardrail typology, detailed information regarding its conservation state, thus providing a more complete and useful framework for inspection and maintenance activities.

3.7.1 Corrosion Level Assessment

The deterioration assessment module operates downstream of the semantic segmentation task. Once the pixel region belonging to the guardrail (ROI) has been identified (Guardrail Isolation explained in section 3.6), the algorithm performs a targeted chromatic analysis to quantify the presence of oxidation, while filtering out false positives caused by reflective elements.
The procedure is articulated in the following phases:

Pre-processing and ROI Isolation

The system input consists of the original image in the BGR color space and the binary mask obtained from the neural network (U-Net). To ensure that the chromatic analysis remains unaffected by the surrounding environment (asphalt, vegetation, sky), a *masking* operation is applied, setting all pixels not belonging to the "guardrail" class to zero.

Color Space Transformation and Normalization

To effectively distinguish rust, the RGB/BGR color space lacks robustness against natural illumination variations (e.g., shadows, overexposure). Consequently, the image is converted into the **HSV** (Hue, Saturation, Value) space, which decouples chromatic information (H) from light intensity (V).
To further mitigate non-uniform lighting conditions along the barrier, the **CLAHE** (*Contrast Limited Adaptive Histogram Equalization*) algorithm is applied exclusively to the **V (Value)** channel.

- **Rationale:** Adaptive equalization enhances local contrast, allowing for the detection of color variations typical of rust even in shadowed or partially obscured areas of the guardrail, thereby increasing the algorithm's robustness in *in-the-wild* scenarios.

Chromatic Segmentation of Rust

The identification of oxidized areas is achieved via *thresholding* within the equalized HSV space. The parameters were empirically defined to isolate the orange-brown hues characteristic of corroded metal:

- **Hue:** Interval $[10, 25]$ (orange/brown spectrum).
- **Saturation:** $S > 40$, to exclude gray areas or dirty but healthy metal.
- **Value:** $V > 40$, to avoid classifying excessively dark or black areas as rust.

Filtering False Positives (Reflectors)

A specific critical issue in guardrail analysis is the presence of bright red retro-reflective delineators. Under certain lighting conditions, the edges of these elements may fall within the rust color range or generate artifacts. To eliminate these false positives, an exclusion logic based on saturation differences and color purity was implemented:

1. A specific mask is generated for **high-saturation red tones** (typical of artificial reflective materials), utilizing the original (non-equalized) HSV image to preserve pure color fidelity.
2. A morphological *opening* operation is applied to remove point noise.
3. The final rust mask (M_{final}) is obtained via a Boolean operation:

$$M_{final} = M_{candidate} \wedge (\neg M_{reflector}) \tag{3.6}$$

   In this manner, only pixels identified as rust that do *not* belong to a reflector are accepted.

Quantification and Classification

The health status of the guardrail is quantitatively defined by the **Rust Ratio** (R_r), calculated as the ratio between the pixels classified as rust (N_{rust}) and the total pixels of the guardrail (N_{total}):

$$R_r = \frac{N_{rust}}{N_{total}} \tag{3.7}$$

To avoid evaluations based on insignificant data (e.g., guardrails that are too distant or partially occluded), the system discards instances where N_{total} falls below a minimum threshold ($N_{min} = 500$ pixels). Based on the R_r value, the guardrail is classified into four severity levels, defined to support maintenance decision-making (Table 3.2).

Level	**Label**	**Threshold (R_r)**	**Description**
0	OK	< 0.01	Absence of corrosion or negligible traces.
1	Warning	$0.01 \leq R_r < 0.05$	Localized corrosion, onset of deterioration.
2	Deteriorated	$0.05 \leq R_r < 0.15$	Significant rust portions, evident degradation.
3	Critical	≥ 0.15	Extensive rust, probable need for replacement.

Table 3.2: Corrosion severity levels based on the Rust Ratio (R_r).

3.7.2 Deformation Level Assessment

The proposed deformation assessment module aims at detecting local misalignments of longitudinal guardrails directly from monocular imagery. The procedure operates entirely in the **image domain** and is therefore inherently linked to the **image resolution and viewing geometry**. Rather than estimating the physical displacement in centimetres, the algorithm produces a *relative deformation index* that can be used to rapidly screen potentially damaged segments along long road stretches.
Given a binary mask produced by the segmentation stage, the method proceeds in four main steps:

(i) Extraction of the guardrail **central axis**,

(ii) Automatic selection of the **proximate portion** of the profile (near the camera),

(iii) Fitting of an **idealised reference model**,

(iv) Computation of a **dimensionless deformation metric** based on the residuals.

Central Axis Extraction from Binary Mask

Let $M(u, v)$ denote the binary mask associated with a single guardrail region-of-interest (ROI) defined over the discrete image grid $\Omega \subset \mathbb{Z}^2$, with $M(u, v) = 1$ for guardrail pixels and $M(u, v) = 0$ for background. For each image column u, we collect the vertical coordinates of all foreground pixels

$$\mathcal{V}(u) = \{v \mid M(u, v) = 1\}. \tag{3.8}$$

If the foreground support is sufficiently populated (i.e., $|\mathcal{V}(u)| \geq N_{\min}$), the corresponding point on the guardrail axis is obtained as the vertical centroid:

$$\hat{v}(u) = \frac{1}{|\mathcal{V}(u)|} \sum_{v \in \mathcal{V}(u)} v. \tag{3.9}$$

This column-wise averaging acts as a low-pass filter on the guardrail geometry, attenuating local irregularities and segmentation noise. The set of axis points is thus given by

$$P_{\text{axis}} = \{(u_i, \hat{v}(u_i))\}_{i=1}^{N}, \tag{3.10}$$

with N denoting the number of valid columns.

Selection of the Proximal Segment

Due to the perspective projection, the apparent size of the guardrail pixels varies significantly along its length: pixels corresponding to the far end of the ROI cover larger real-world areas and are more affected by discretisation noise. To mitigate this effect and work under an **approximately constant ground sample distance (GSD)**, the deformation analysis is intentionally **restricted to the *proximal* portion** of the guardrail, i.e., the segment closest to the camera.

Let $\{(u_i, \hat{v}_i)\}_{i=1}^{N}$ be the axis points sorted by their horizontal coordinate u_i. Since the image coordinate system has its origin in the top-left corner, larger v values correspond to points closer to the camera. We therefore estimate which side of the profile (left or right in image space) is closer by comparing the mean v value over

small windows at the beginning and at the end of the profile:

$$\bar{v}_{\text{start}} = \frac{1}{K} \sum_{i=1}^{K} \hat{v}_i, \tag{3.11}$$

$$\bar{v}_{\text{end}} = \frac{1}{K} \sum_{i=N-K+1}^{N} \hat{v}_i, \tag{3.12}$$

with K set as a fraction of N and bounded below by a minimum number of points. The side with the larger mean value is considered the proximal side.
From this proximal side, we then retain only a fixed fraction α (e.g., $\alpha = 0.3$) of the profile:

$$P_{\text{near}} = \{(u_i, \hat{v}_i)\}_{i \in \mathcal{I}_{\text{near}}}, \tag{3.13}$$

where $\mathcal{I}_{\text{near}}$ indexes either the first or the last αN samples, depending on which side is closer. In subsequent frames of the same video sequence, the same rule is applied, so that the analysis window always corresponds to the portion of guardrail that is spatially close to the vehicle. When the frames are temporally adjacent and the trajectory is smooth, this strategy ensures that the GSD within the analysed segment remains approximately constant along the sequence, even though the exact same physical points are not explicitly tracked.

Parametric Fitting of the Idealised Profile

The proximal axis points P_{near} are assumed to represent the superposition of the global road curvature and local deformations caused by impacts or structural imperfections. To separate these contributions, the ideal (undamaged) trajectory is approximated by a **low-order polynomial model**:

$$y_{\text{model}}(u) = \sum_{j=0}^{k} \beta_j u^j, \tag{3.14}$$

where k is typically chosen equal to 1 (straight alignment) or 2 (mild curvature). The parameter vector $\boldsymbol{\beta}$ is estimated via Ordinary Least Squares (OLS), minimising the

squared residuals between the observed axis and the model:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{(u_i,\hat{v}_i)\in P_{\text{near}}} \left(\hat{v}_i - \sum_{j=0}^{k} \beta_j u_i^j \right)^2 . \tag{3.15}$$

Because of its limited flexibility, the polynomial model effectively captures only the slowly varying component of the geometry (road alignment), while sharp local deviations are left in the residuals and interpreted as potential damage.

Relative Deformation Metric

For each sample $(u_i, \hat{v}_i) \in P_{\text{near}}$, the **vertical residual** is defined as

$$r_i = \hat{v}_i - y_{\text{model}}(u_i), \tag{3.16}$$

and we consider its absolute magnitude $|r_i|$ as an indicator of local bending. To obtain a deformation metric that is robust to perspective scaling and independent of the absolute pixel values, the residuals are normalised by the apparent vertical extent of the guardrail in the mask:

$$h_{\text{rail}} = v_{\max} - v_{\min}, \quad v_{\max} = \max\{v \mid M(u,v) = 1\}, \quad v_{\min} = \min\{v \mid M(u,v) = 1\}. \tag{3.17}$$

The *relative deviation* is then defined as

$$D_{\text{rel}}(u_i) = \frac{|r_i|}{h_{\text{rail}}}. \tag{3.18}$$

From this profile, summary statistics such as the maximum and mean relative deviation are computed:

$$D_{\max}^{\text{rel}} = \max_i D_{\text{rel}}(u_i), \tag{3.19}$$

$$D_{\text{mean}}^{\text{rel}} = \frac{1}{|P_{\text{near}}|} \sum_i D_{\text{rel}}(u_i). \tag{3.20}$$

In the implementation used for this work, the **binary decision "deformed / not deformed"** for a given ROI and frame is taken by thresholding the maximum relative

deviation:

$$\text{deformed} = \begin{cases} 1 & \text{if } D_{\max}^{\text{rel}} \geq T_{\text{def}}, \\ 0 & \text{otherwise}, \end{cases} \tag{3.21}$$

with $T_{\text{def}} = 0.02$. Values above this threshold indicate that, at least in one column of the analysed segment, the guardrail axis exhibits a deviation larger than 2% of its apparent height.

It is important to stress that **this index is *dimensionless* and image-based**: the algorithm does not attempt to infer the true displacement in centimetres, nor does it rely on external calibration. Instead, it provides a consistent, resolution-dependent indicator that can be used to quickly identify segments where the geometry deviates significantly from an ideal alignment and which therefore require closer inspection.

3.8 Geospatial Database Integration

The final step in the pipeline consists of generating the monitoring database using all previously obtained results (see Figure 3.12). This stage integrates the outputs of the segmentation, classification, and condition-assessment modules into a unified geo-referenced inventory of roadside guardrails.

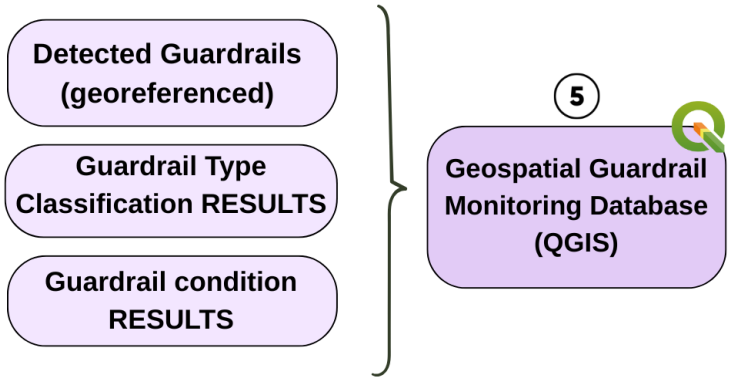


Figure 3.12: Fifth block of the pipeline: Geospatial Guardrail Monitoring Database implementation.

To enable the geospatial visualization and analysis of the guardrail information extracted from the video recordings, a **dedicated spatial database** was created. This database acts as the bridge between the automated processing pipeline and its

interpretation within a **Geographic Information System (GIS)**, enabling the results to be explored, queried, and contextualized on the road network. To achieve this, we carried out two main phases:

1. Preparation of a **structured CSV file** containing georeferenced guardrail annotations
2. Conversion of this file into an editable and queryable **GeoPackage database**.

3.8.1 Guardrail Data Structuring

For each guardrail instance, a record was generated containing the following attributes:

- **Frame name**
- **Geographic coordinates**: latitude, longitude, height- obtained through the the temporal synchronization of frame timestamps with GNSS measurements;
- **Guardrail type** (double-wave/triple-wave/bridge parapet barrier/central median barrier);
- **Deterioration status** (OK/To be monitored/ Deteriorated/Critical);
- **Deformation status** (True= deformed, False= intact);
- **Source of acquisition** (e.g., Politecnico di Torino);
- **Acquisition date**, enabling temporal tracking and historical comparison.

The resulting file thus acts as a clean and structured tabular dataset, containing all metadata associated with each detected guardrail segment.

3.8.2 Spatial Database Format

To ensure portability, consistency, and efficient querying, the guardrail dataset is stored in a spatial database format compatible with common GIS tools. In this work, the *GeoPackage (GPKG)* format is adopted due to its ability to:

- store spatial geometries and attribute tables within a single file;

- support SQL-based querying and indexing;
- ensure interoperability with most GIS environments, such as QGIS;
- provide efficient access and editing capabilities.

The GeoPackage structure allows the entire guardrail inventory to be represented as a single layer whose attributes encode both the semantic information (type, condition) and the associated geospatial metadata. This enables a range of analysis tasks, including:

- **Geospatial visualization** of guardrail points along the surveyed road segment;
- **Attribute-based queries**, for example filtering by guardrail type or condition;
- **Combination with additional layers**, including orthophotos, road networks, or administrative boundaries;
- **Production of thematic maps** for reporting and inspection purposes.

This integration provides the final step of the methodological pipeline, enabling the automated results to be inspected, validated, and interpreted with respect to the real-world geographic context.

Chapter 4

Case Study: Application of the Methodology, and Results

This chapter presents the practical application of the proposed methodology to the selected case study, with the aim of evaluating the effectiveness of the developed approach for the identification and analysis of roadside guardrails along a real road segment. After outlining the methodological pipeline, the datasets, and the supporting tools, this chapter provides a detailed description of all implemented steps—from data acquisition and preprocessing to guardrail segmentation, classification, and condition assessment.

Particular attention is devoted to the results obtained at each stage of the pipeline, including georeferencing procedures, structural condition analysis, and GIS-based integration, in order to provide a comprehensive evaluation of the overall performance of the proposed system.

The objectives of this chapter are therefore twofold:

- to demonstrate **how the methodology has been applied in a real-world scenario**;
- to critically **discuss the achieved results**, highlighting strengths and limitations.

4.1 Case Study Description

This case study focuses on a specific road segment located within the metropolitan area of Turin (Italy). The selected area corresponds to a highway route personally surveyed and recorded by us using a vehicle-mounted multi-sensor platform (described in section 3.3. The recorded route spans several kilometres and includes both straight and curved sections, multiple lanes, ramps, and roadside vegetation, providing natural variability in illumination, perspective, and occlusions. This makes it a suitable real-world environment for validating the proposed methodology for guardrail detection, segmentation, and condition assessment.
The acquired dataset was intentionally divided into two parts according to the objectives of the study:

- **Training Section:** the first portion of the route was used to generate ground truth annotations and train/validate the neural segmentation models.
- **Testing Section:** the final 2.3 km of the recorded highway segment were reserved for evaluating the performance of the end-to-end methodology, including detection, segmentation, geolocation and structural assessment of guardrail elements.

A map visualizing the surveyed route, highlighting both the training and testing portions, is included to provide spatial context for the case study (see Figure 4.1). This geospatial representation allows for a clear understanding of where the dataset was collected and where the methodology was validated.

Figure 4.1: Reconstructed trajectory.

Overall, this case study offers a controlled yet realistic scenario for assessing the robustness of the proposed pipeline and its applicability to large-scale roadside infrastructure monitoring.

4.2 Data Preparation and Synchronization

4.2.1 GPS post-processing

The processing procedure was carried out using the `RTKPost` module of `RTKLib`, which allows the integration of rover and base station data to obtain higher-accuracy positioning solutions. The positioning mode was set to *Kinematic*, which is suitable for scenarios involving a moving receiver, such as the vehicle-based survey. Both *L1* and *L2* GNSS frequencies were employed in a combined solution to enhance the robustness and reliability of the estimated trajectory. The system utilized both *GPS* and *GLONASS* constellations, increasing the number of satellites available for computation and enhancing overall coverage.
Integer ambiguity resolution was managed using the *Fix and Hold* strategy, which maintains a fixed solution once a sufficient level of reliability has been reached.
The reconstructed trajectory can be visualized through `RTKLib`'s integrated graphical tool.

Figure 4.2: Reconstructed trajectory using `RTKLib`. Fix quality is color-coded.

As shown in Figure 4.2, each point in the trajectory is color-coded according to the fix quality indicator Q:

- **Green**: fixed solution ($Q = 1$), with centimeter-level accuracy
- **Yellow**: float solution ($Q = 2$), with sub-meter accuracy

Out of a total of 1632 points, 1040 (63.7%) were fixed and 592 (36.3%) were float. The predominance of fixed solutions indicates a high level of reliability in the reconstructed trajectory, making it suitable for downstream tasks such as video frame synchronization and spatial interpolation.

4.2.2 Frame extraction strategy

As mentioned in Section 4.x, to avoid generating an excessive number of redundant images while ensuring adequate coverage of the road environment, an adaptive frame sampling strategy was developed based on the instantaneous speed of the vehicle, which was calculated from the previously obtained `.pos` file.
Figure 4.3 shows the speed profile. The analysis reveals several phases during which the speed remains very low (below 10 km/h), alternating with segments where the speed exceeds 50 km/h, occasionally reaching peaks above 65 km/h. At times, the vehicle comes to a complete stop, whereas in others, the speed varies significantly within a few seconds.

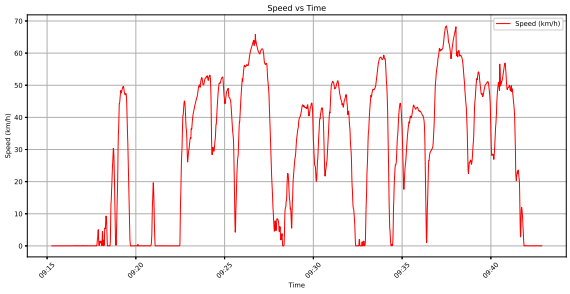


Figure 4.3: Speed profile of the vehicle during the survey.

For this reason, an adaptive sampling strategy was implemented for frame extraction from video recordings. Specifically, three speed thresholds were defined:

- Speed < 50 km/h $\rightarrow$ 1 frame per second
- 50 km/h $<$ speed < 65 km/h $\rightarrow$ 3 frames per second
- Speed > 65 km/h $\rightarrow$ 5 frames per second

The rationale behind this approach is that, at higher speeds, the scene evolves more quickly in space. Therefore, to avoid losing relevant visual information (such as the presence or absence of guardrails), it is essential to increase the sampling rate. Conversely, during low-speed or stationary conditions, the scene is relatively static, and fewer frames are sufficient to represent it.

4.2.3 Georeferencing frames

At this stage, it was necessary to assign a geographic position to each frame extracted from the video, ensuring consistency with the vehicle's actual trajectory. Since the video frames were sampled at a variable frequency, their timestamps did not always match those available in the `.pos` file. To ensure accurate georeferencing, a linear time interpolation was applied to the GPS data, as described in the subsection 3.4.4.

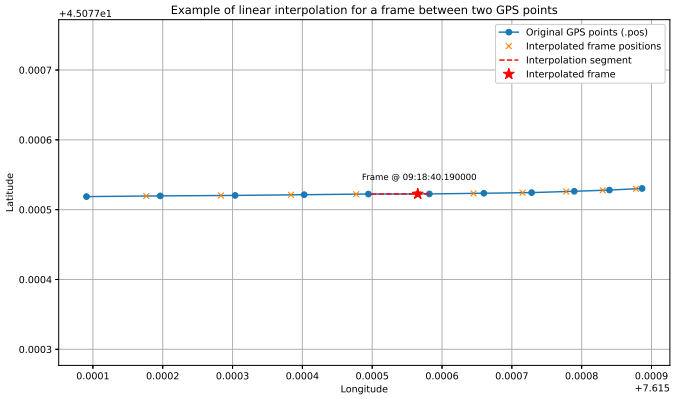


Figure 4.4: Example of frame georeferencing via linear interpolation between two GPS points

As an illustrative example, Figure 4.4 shows a concrete case of interpolation applied to a video frame located between two consecutive GPS points. The blue circles represent the original GPS positions extracted from the `.pos` file, each associated with an exact second, while the crosses indicate the interpolated frames corresponding to intermediate timestamps. The red dashed line connects two consecutive GPS points and temporally bounds the highlighted frame, clearly illustrating the estimated position.

In the specific example shown, the GPS receiver recorded a position at 09:18:40.000,

and the next one at 09:18:41.000. However, one of the extracted frames corresponds to the timestamp 09:18:40.190, an intermediate instant not explicitly represented in the GNSS data. The figure visually demonstrates the outcome of the interpolation: the frame is correctly placed along the segment connecting the two GPS points, confirming the effectiveness of the adopted approach.

4.3 Guardrail Detection

In this section, we present the quantitative and qualitative results obtained from training the U-Net model described in Chapter 3.5.

4.3.1 Dataset and Preprocessing

A total of approximately 2000 RGB frames (512×512) were extracted from the video recorded by the left stereo camera mounted on the vehicle. Although 1400 images were initially assigned to the training set, the network does not benefit from training on long sequences of highly similar frames. Instead, what is required is sufficient scene variability. For this reason, only about 150 frames were manually annotated to obtain pixel-level ground truth masks. Figure 4.5 shows examples of manually annotated masks through the `Image Labeler` tool of Matlab.

Figure 4.5: Examples of manually annotated guardrail masks created using Matlab's `Image Labeler` tool.

To further increase diversity and improve generalization capability, additional images and masks were included from the `Mapillary dataset`: the original dataset was

filtered to retain only scenes containing European guardrails, and the corresponding masks were derived from publicly available JSON annotation files. Figure 4.6 shows an example of an image-mask pair obtained from Mapillary.

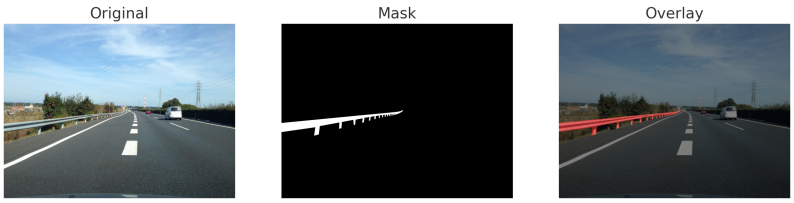


Figure 4.6: Example of an original road image, its corresponding guardrail mask, and the resulting overlay produced from the Mapillary dataset.

All images were normalized to the range [0,1], and all masks were binarized. The final dataset used for training consisted of a random split of 80% for training and 20% for validation.

4.3.2 U-Net Training and Parameters

The model was implemented in *TensorFlow/Keras*. Based on preliminary tuning, the final training configuration was established as follows: the encoder consists of **four downsampling stages** with 64, 128, 256, and 512 filters, each consisting of **two 3×3 convolutions** followed by **max pooling and dropout** (0.3). The bottleneck employs **1024 filters**. The decoder reconstructs the original spatial resolution using **transposed convolutions** and **skip connections** with the corresponding encoder feature maps. A final 1×1 convolution with **sigmoid activation** produces the binary guardrail mask.

The model was trained for **40 epochs** using the **Adam optimizer** with a **learning rate of** 1×10^{-4}. Due to the severe class imbalance — guardrail pixels make up only a very small portion of each frame — the **Binary Focal Cross-Entropy loss** (with $\gamma = 2.0$) was adopted. Early stopping and adaptive learning-rate reduction were applied to mitigate overfitting. The final model was selected based on the best validation **Binary IoU**. Table 4.1 summarizes the adopted configuration and the hyperparameters used for training.

Component	Configuration
Framework	TensorFlow / Keras
Hardware	NVIDIA T4 GPU
Encoder	4 downsampling stages (64, 128, 256, 512 filters)
Encoder blocks	Two 3×3 convolutions per stage
Pooling	Max pooling (per stage)
Dropout	0.3 after each downsampling stage
Bottleneck	1024 filters
Decoder	Transposed convolutions + skip connections
Output layer	1×1 convolution + sigmoid activation
Training epochs	40
Optimizer	Adam
Learning rate	1×10^{-4}
Loss function	Binary Focal Cross-Entropy ($\gamma = 2.0$)
Class imbalance handling	Focal loss
Regularization	Early stopping, ReduceLROnPlateau
Selection criterion	Best validation Binary IoU

Table 4.1: Summary of model architecture and training configuration.

4.3.3 Training Performance Evaluation

Model performance was evaluated using standard semantic segmentation metrics, including Intersection-over-Union (IoU), Dice coefficient, Precision, Recall, and the area under the Precision–Recall curve (PR-AUC). The training curves of the loss and Dice coefficient are reported in Figure4.7, showing stable convergence with a consistent decrease in training loss and a corresponding improvement in segmentation accuracy. The training loss decreases continuously, while the validation loss follows a similar trend until reaching a plateau phase with no signs of divergence, indicating good generalization. Similarly, the Dice coefficient increases rapidly during the initial epochs and then converges to high values (approximately 0.85–0.90), with comparable trajectories for both training and validation. This behaviour confirms that the model effectively learned the shape and extent of guardrails without overfitting the dataset.

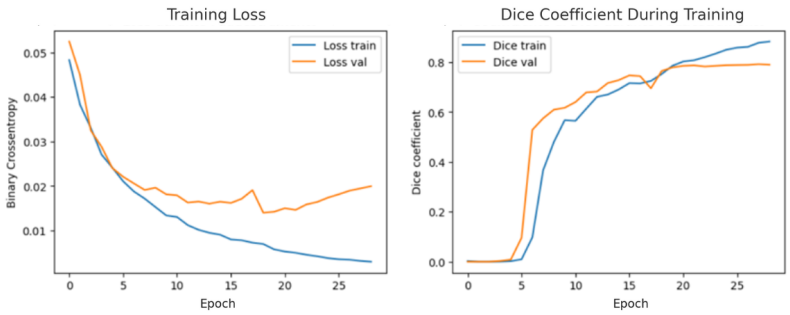


Figure 4.7: Training and Validation curves. Left: Binary Focal Loss reduction over epochs. Right: Improvement of the Dice Coefficient (thresholded at 0.5) on the validation set.

4.3.4 Quantitative Evaluation

The model performance was also evaluated on a held-out test set of 19 images using pixel-wise metrics. A threshold of 0.5 was applied to the sigmoid output to generate the final binary masks.

Metric	Score
Mean IoU	0.8017
Mean Dice	0.8866
Precision@0.5	0.8855
Recall@0.5	0.9018
PR-AUC	0.9712

Table 4.2: Summary of segmentation performance on a test dataset.

Figure 4.8 and Figure 4.9 illustrate the overall distribution of performance metrics over the test set. The Precision–Recall curve shows that the classifier maintains high precision for almost all recall values, confirming its robustness. The IoU histogram demonstrates that most images achieve values between 0.80 and 0.90, indicating consistent segmentation quality. Only a small number of frames fall below 0.70, generally corresponding to challenging scenarios where the guardrail is either far from the camera, appears very thin in the image, or is partially occluded.

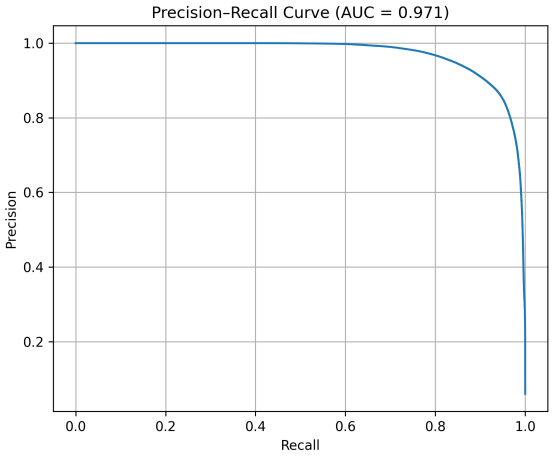


Figure 4.8: Precision–Recall curve for the guardrail segmentation model. The high AUC (0.9712) indicates strong discriminative ability between guardrail and background pixels.

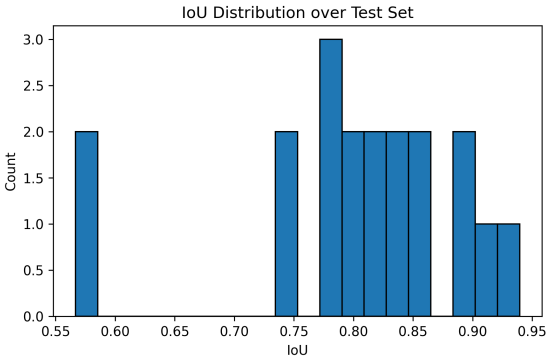


Figure 4.9: Distribution of IoU values over the test set. Most predictions fall between 0.80 and 0.90.

4.3.5 Qualitative Analysis and Error Modes

Visual inspection of the predictions reveals the model's strengths and limitations. Figure 4.13 shows three representative examples from the test set, including some of the most challenging cases. These images directly correspond to those discussed in the text regarding robustness to curvature, occlusions, and typical error modes.

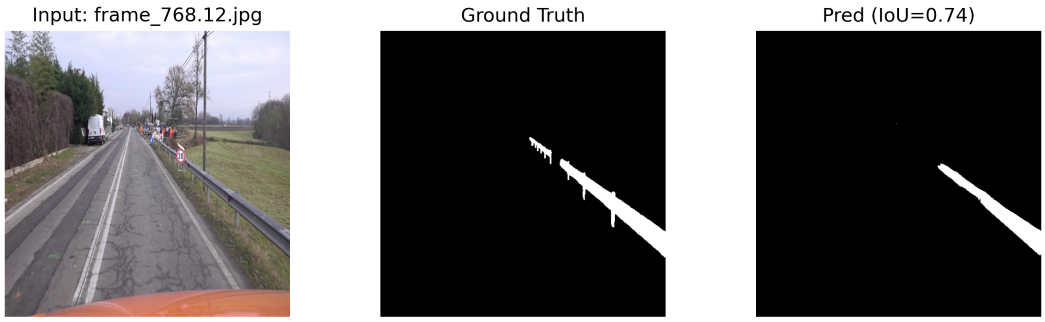


Figure 4.10: Example of segmentation with IoU=0.74.

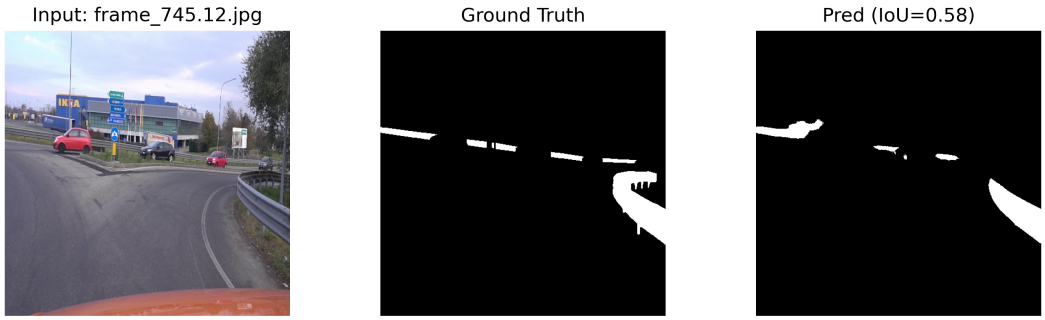


Figure 4.11: Example of segmentation with IoU=0.58.

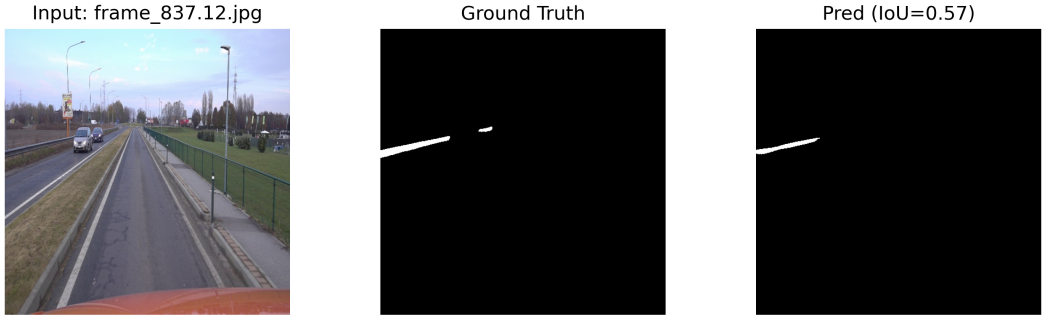


Figure 4.12: Example of segmentation with IoU=0.57.

Figure 4.13: Qualitative examples showing input images, ground truth masks, and corresponding predictions for some of the most challenging frames in the test set.

4.3.6 Application to the Final Road Segment

The trained U-Net was finally applied to the last portion of the road used in this study, which was not included in either the training or validation phases. This final segment represents the stretch where the complete inspection methodology is ultimately deployed. The results are overall positive: the network reliably recognizes the presence of guardrails along the entire route and accurately localizes them within each frame. Only in a few isolated cases does the model fail to detect distant guardrails, whose visual footprint is extremely small due to perspective. This limitation is not particularly problematic, as these far objects naturally appear larger and more detectable in subsequent frames of the same sequence. Overall, this confirms that the model generalizes well and is suitable for supporting the downstream analysis pipeline adopted in the final road inspection.

4.4 Guardrail classification

As described in the methodology chapter, the task of guardrail typology classification is performed using a ResNet-18 convolutional neural network. The network was trained on a dataset derived from the same frames used to train the U-Net segmentation model. From each annotated frame, the *regions of interest* (ROI) containing the guardrail segments were extracted by applying the binary segmentation masks. Whenever multiple guardrails appeared within the same frame (e.g., left and right sides), each ROI was treated as an independent sample, increasing intra-class variability and improving the robustness of the classifier.
Once trained, the ResNet-18 model was used to classify the ROIs extracted from the output of the U-Net segmentation network applied to the last segment of the road corridor considered in this study.

4.4.1 ResNet-18 Training and Parameters

To enhance model robustness under real-world acquisition conditions (variable illumination, reflections, shadows, perspective distortions), the following **data augmentation strategies** were applied to the ROI dataset:

- **Resize** to 224×224 pixels (standard ResNet input format)

- **Random horizontal flip** with probability $p = 0.5$
- **Color jitter** on brightness, contrast and saturation ($\pm 20\%$)
- **Normalization** using ImageNet statistics

These augmentations were essential to prevent overfitting, given the relatively small number of samples per class. The dataset was then partitioned using a standard hold-out strategy: 80% **for training** and 20% **for validation,** organized using a folder-based class structure.
The model was trained for 20 **epochs** using the hyperparameters in Table 4.3.

Parameter	**Value**
Batch size	16
Optimizer	Adam (fast convergence)
Learning rate	1×10^{-4}
Loss function	CrossEntropyLoss
Hardware	NVIDIA T4 GPU

Table 4.3: Hyperparameters used for training the ResNet-18 classifier.

The relatively low learning rate was selected to preserve the knowledge encoded in the pre-trained weights and mitigate the risk of catastrophic forgetting. During training, the evolution of both loss and accuracy was monitored for the training and validation sets across all epochs to detect potential instability or overfitting.

4.4.2 Training Results

After the first epoch, the network already achieves a training accuracy of 78.30% and a validation accuracy of 89.66%, together with a consistent drop in the loss (from 0.6713 to approximately 0.2609 on the validation set). By the final epoch, the model reaches a **training accuracy of** 99.53% and a **validation accuracy of** 98.28%, with the **validation loss** remaining stable in the range 0.02–0.07**.** The absence of divergence between the training and validation curves suggests that the model is not overfitting despite the limited dataset size, and is effectively learning discriminative features relevant for guardrail typology classification.

On the validation set, consisting of 58 samples, **the model misclassifies only one instance**.

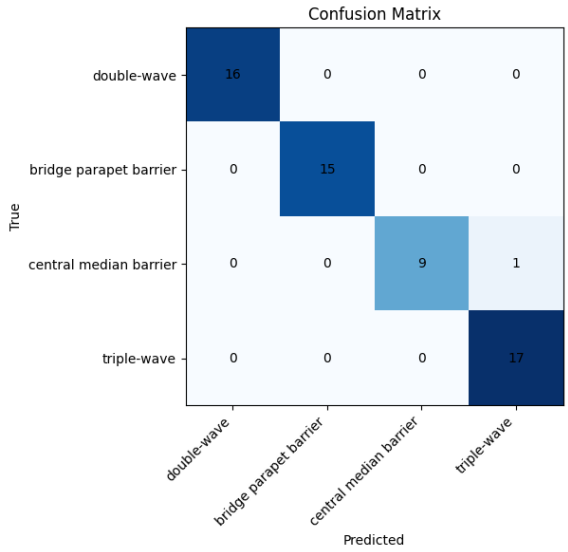


Figure 4.14: Confusion matrix of the ResNet-18 guardrail classifier on the validation set.

The confusion matrix in Figure 4.14 illustrates the classification performance across the four classes:

- **double-wave**: 16 correct out of 16
- **bridge parapet barrier**: 15 correct out of 15
- **central median barrier**: 9 correct out of 10
- **triple wave**: 17 correct out of 17

The single misclassification corresponds to a **central median barrier** guardrail predicted as **triple-wave**(Figure 4.15), a mistake that can be explained by the morphological similarity of the two profiles and the presence of vegetation partially occluding the discriminative region. Despite this isolated error, the macro-average F1-score of 0.98 confirms the reliability of the model in distinguishing among the different guardrail typologies included in the dataset.

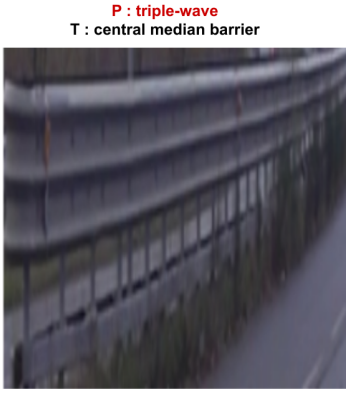


Figure 4.15: Example of the only misclassified sample: a `central median barrier` guardrail incorrectly predicted as `triple-wave`.

4.4.3 Qualitative Analysis of the Trained ResNet-18 on the Real Road Segment

The trained ResNet-18 classifier was applied to all *Regions of Interest* (ROIs) extracted by the U-Net segmentation model on the final portion of the motorway under analysis. In total, the model processed **864 ROIs**, each corresponding to an individual guardrail instance. These results therefore reflect the inference-time behaviour of the classifier on data never seen during training.

The overall distribution of predicted classes is reported in Table 4.4. This distribution is not incidental but aligns with the actual infrastructure present along the analysed road segment: **triple-wave** guardrails and **bridge parapet barriers** are indeed the most common configurations in the final section of the route. The substantial presence of the **central median barrier** class is also coherent with the central divider found along long portions of the motorway. Conversely, the **double-wave** category is less represented, both in reality and in the model predictions.

Predicted class	Occurrences
triple-wave	315
bridge parapet barrier	297
central median barrier	224
double-wave	28
Total	864

Table 4.4: Distribution of guardrail typologies predicted by the ResNet-18 classifier on the real road segment.

Although no manually annotated ground truth is available for this segment of the road, the classifier exhibits overall stable behaviour.

Temporal Coherence Analysis

To further assess the operational stability of the model, a **temporal coherence analysis** was carried out by examining the predicted class for each ROI across consecutive frames. Ideally, if the same guardrail remains visible at a given image position, the predicted class should remain stable over time unless an actual change in infrastructure occurs.

The analysis reveals a total of **only 12 class transitions** between consecutive frames, distributed across both `roi_0` and `roi_1` (typically `roi_0` for the right side, `roi_1` for the left side). These transitions do not constitute systematic errors but rather reflect local variability in the predictions across closely spaced frames. Most fluctuations occur between morphologically similar classes such as **triple-wave**, **bridge parapet barrier**, and **central median barrier**. These structures can appear visually similar when observed from unfavourable viewpoints, partially occluded by vegetation, traffic or additional guardrails.

Table 4.5 reports representative examples of such transitions. These oscillations are not due to an actual change in the physical guardrail but are likely caused by small geometric variations in the ROIs extracted from the segmentation masks.

ROI	Previous (frame / class)	Next (frame / class)
0	1284.92 / `triple-wave`	1285.12 / `bridge parapet`
0	1297.72 / `bridge parapet`	1297.92 / `triple-wave`
1	1304.35 / `triple-wave`	1304.56 / `median barrier`
1	1321.12 / `median barrier`	1321.35 / `triple-wave`

Table 4.5: Examples of class transitions across consecutive frames (compact version).

Overall, the inference results confirm that the classifier is capable of correctly identifying the predominant guardrail typologies in real-world operational settings. The small number of localised and non-systematic fluctuations does not compromise the global stability of the model. This behaviour supports the robustness of the proposed segmentation–classification pipeline and demonstrates its suitability for large-scale, real-time monitoring of roadside barriers.

4.5 Corrosion Level Assessment

The corrosion analysis algorithm was applied to all ROIs generated by the U-Net segmentation model for the final portion of the highway included in the case study. For each ROI, the corresponding binary mask was first denoised to remove small artifacts and spurious pixels, ensuring that only the actual guardrail surface contributed to the deterioration estimation. As defined in the Methodology chapter, ROIs with an area smaller than $N_{\min} = 500$ pixels were excluded from the analysis. This threshold guarantees sufficient statistical reliability and prevents false detections caused by segmentation fragmentation (for instance, when the guardrail is partially occluded by vehicles or vegetation). Applying this filter led to the exclusion of 6 instances, which were labeled as *Not evaluable*.

4.5.1 Analysis of the Guardrail Status: Results

Table 4.6 summarizes the classification results for the valid samples ($N_{\text{valid}} = 835$), categorized according to their deterioration level.

Condition	Count	Percentage (%)
Good	812	97.3%
To be monitored	21	2.4%
Deteriorated	0	0.0%
Critical	2	0.2%

Table 4.6: Classification results on the test sequence.

The results reveal a clear predominance of guardrails in **good condition** (97.3%), which is expected in a highway segment subject to regular maintenance activities. Nevertheless, the system successfully detected specific areas requiring attention, demonstrating sensitivity to varying degrees of oxidation, including localized rust indicative of early deterioration.
Figure 4.16 provides a representative example of rust detection within a single guardrail.

Figure 4.16: Example of detected rust sign within a single guardrail.

A relevant observation concerns the presence of retroreflective elements. As shown in Figure 4.17, these components do not introduce systematic errors in the analysis, as the deterioration metric relies on textural and chromatic features typical of corrosion rather than brightness spikes produced by reflective surfaces.

Figure 4.17: Example of how retroreflectors don't affect corrosion estimation.

4.5.2 Additional Analysis on Paint Loss

In an early phase of the study, the possibility of incorporating a dedicated ***paint loss measure*** into the deterioration assessment was explored. The idea was to combine rust detection with changes in grayscale intensity to capture surface opacity or abrasion.

In some cases, this method proved effective, as shown in Figure 4.18, where genuine paint degradation is correctly highlighted. However, further experiments revealed that the method is highly sensitive to illumination variability. Shadows, sunlight reflections, and exposure differences along the road produce significant false positives: guardrails in perfect condition may be incorrectly classified as degraded, as illustrated in Figure 4.19.

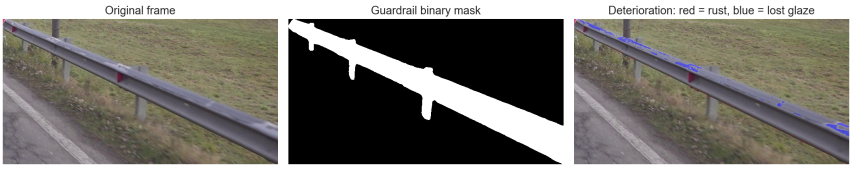


Figure 4.18: Example of true paint loss correctly identified.

Figure 4.19: Example of true paint loss wrongly identified: false positive due to lighting conditions.

Due to this **strong dependence on lighting conditions** and its **limited robustness** in real-world outdoor scenarios, the paint-loss criterion was not included in the final pipeline. This choice improved the stability and reliability of the overall system, which ultimately relies solely on texture- and color-based rust detection, a more consistent indicator of structural deterioration.

4.6 Deformation Assessment

The **deformation assessment methodology** described in the previous chapter was applied to all guardrail regions-of-interest (ROIs) extracted along the selected highway segment. For every frame, the binary masks generated by the segmentation model were processed to extract the axis, identify the proximal segment, and compute the relative deformation metrics.

4.6.1 Results on representative frames

Before analysing the full temporal sequence, it is useful to examine a few representative frames individually. These examples illustrate how the deformation index reacts to the presence or absence of geometric anomalies in the guardrail.
Figure 4.20 shows a frame containing a clearly **deformed** guardrail segment. The bending is visually detectable in the RGB image and is faithfully captured by the extracted axis profile: the fitted line diverges from the real centroidal curve, especially in the initial portion of the guardrail near the camera. This produces a local increase in the relative deviation, well above the threshold of 0.02, and leads to a correct "deformed" classification.

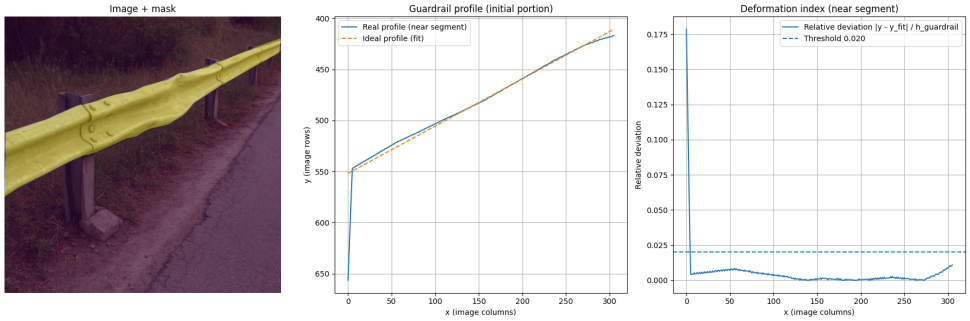


Figure 4.20: Example of a visibly **deformed** guardrail segment, with the corresponding profile and deformation index.

In contrast, Figure 4.21 illustrates a frame where the guardrail is clearly **intact**. The extracted centroidal axis remains almost perfectly aligned with the fitted model, and the relative deviation stays consistently below the threshold over the entire proximal segment. The deformation curve is flat and near zero, confirming the non-deformed condition.

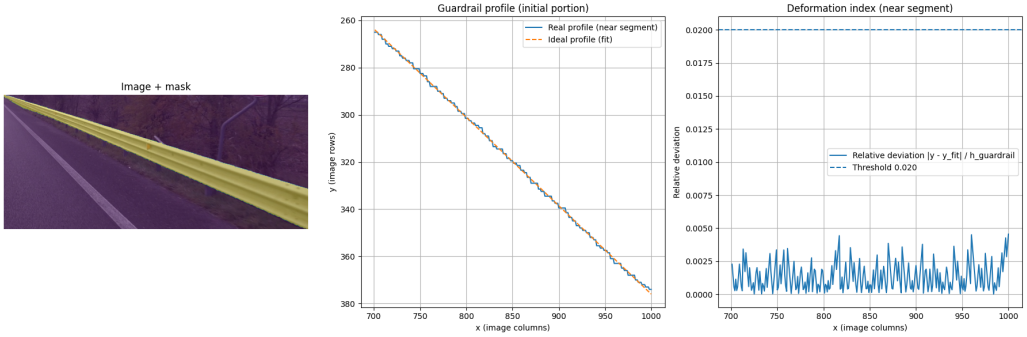


Figure 4.21: Example of an **intact** guardrail segment with minimal deviation from the fitted profile.

These examples demonstrate that the proposed algorithm behaves consistently with visual inspection. When the guardrail exhibits a noticeable geometric inflection, the deviation between the real and fitted profiles increases. Conversely, when the barrier is straight and undamaged, both the axis profile and the deformation index remain stable. This confirms that the method is sensitive to actual geometric anomalies and robust to minor segmentation noise.

4.6.2 Sequence-level analysis and Results

To illustrate the behaviour of the method along a temporal sequence, this section focuses on ROI 0, which corresponds to the right-hand side of the scene.
A total of 201 frames were analysed for ROI 0. Frames classified as non-deformed exhibited consistently low values of $D_{\max}^{\text{rel}}$ (median ≈ 0.011), in agreement with a nearly straight and undamaged barrier. Conversely, frames labelled as deformed showed significantly higher deviations (median ≈ 0.034), with peaks reaching approximately 0.17. Overall, around 60 frames (roughly 30% of the entire sequence) were classified as deformed.
From a temporal viewpoint, the **deformed frames do not appear randomly scattered**. Instead, they form *clusters of consecutive frames*, each corresponding to a physically damaged segment of the guardrail. The most pronounced cluster spans approximately 10-12 consecutive frames and matches a clearly visible lateral bulge in the structure. Outside these clusters, the deformation index remains low, and the guardrail is correctly classified as non-deformed.

These results confirm that the proposed method serves as an effective **screening tool** for the **rapid identification of anomalous geometries** along extended road sections. After processing the full sequence, the operator needs to inspect only the limited subset of frames that exhibit high deformation indices.
Two additional considerations must be explicitly acknowledged:

- **The analysis is qualitative rather than quantitative.** Because the method operates entirely in the image domain, the deformation index does not correspond to a physical displacement. It identifies the *presence* and *relative severity* of damage but does not quantify its magnitude in centimetres.
- **Sequential consistency is ensured via the proximal-segment strategy.** By analysing only the image region closest to the camera, the algorithm maintains an approximately constant GSD across consecutive frames. Due to the natural overlap between frames along the vehicle trajectory, the proximal portion of frame t corresponds to a slightly shifted, but spatially adjacent, portion of the same guardrail in frame $t+1$. This guarantees internal coherence of the deformation measurements across the sequence, even without explicit 3D tracking.

In summary, the deformation module provides a computationally efficient, image-based mechanism for identifying segments of roadside barriers that may require further inspection. While it does not replace a metric structural evaluation, it offers a robust and consistent tool for prioritising areas of potential interest in large-scale visual surveys.

4.7 Integration and Visualization of Guardrail Data in the GIS Environment

This section describes the practical implementation of this process in the case study, detailing the construction of the dataset, its import into QGIS, the conversion to a GeoPackage database, and the subsequent spatial queries and map visualizations.

4.7.1 Construction of the Guardrail CSV Dataset

The first step consisted in generating a structured CSV file containing one record for each detected guardrail region (ROI) in the video sequence. This file was created by merging the outputs of the segmentation, classification, deterioration, and deformation modules with the GNSS-based georeferencing information (see section 3.8) . Figure 4.22 shows an excerpt of the resulting CSV file.

Frame name	latitude	longitude	height	Guardrail type	Deterioration status	Deformation status	Source of acquisition	Acquisition date
frame_1255.32_roi_0.jpg	45.092432	7.581370	329.5796	triple-wave	OK	False	Politecnico di Torino	28/11/2024
frame_1255.32_roi_1.jpg	45.092432	7.581370	329.5796	bridge parapet barrier	OK	False	Politecnico di Torino	28/11/2024
frame_1255.52_roi_0.jpg	45.092410	7.581400	329.5916	triple-wave	OK	False	Politecnico di Torino	28/11/2024
frame_1255.52_roi_1.jpg	45.092410	7.581400	329.5916	bridge parapet barrier	OK	False	Politecnico di Torino	28/11/2024
frame_1255.72_roi_0.jpg	45.092388	7.581430	329.6036	triple-wave	OK	False	Politecnico di Torino	28/11/2024

Figure 4.22: Excerpt of the `.csv` file generated for the case study. Each row corresponds to a detected guardrail region.

4.7.2 From CSV File to GeoPackage Database

The CSV file was imported into QGIS using the *Add Delimited Text Layer* tool. This interface allows the file to be loaded as a temporary point layer, using latitude and longitude fields as geometry attributes. Figure 4.23 illustrates the import dialog used in this step.

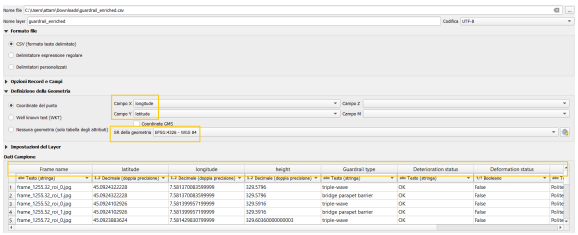


Figure 4.23: QGIS *Add Delimited Text Layer* interface used to import the guardrail CSV dataset.

After confirming the coordinate system (**EPSG:4326**), the layer was correctly visualized on the map canvas as a temporary point layer.
To obtain a persistent and editable spatial database, the temporary CSV layer was exported as a *GeoPackage (GPKG)* file using the *Save Features As* functionality. The resulting database contains all attributes and geometries in a single structured file.

4.7.3 Spatial Visualization of the Guardrail Inventory

Once the GeoPackage database was created, it was loaded into QGIS and visualized on top of the official road cadastral map of the Piedmont region. This reference map, downloaded from the *Geoportale Piemonte*, provides an accurate and up-to-date representation of the regional road network, enabling the spatial context of each detected guardrail to be clearly inspected.

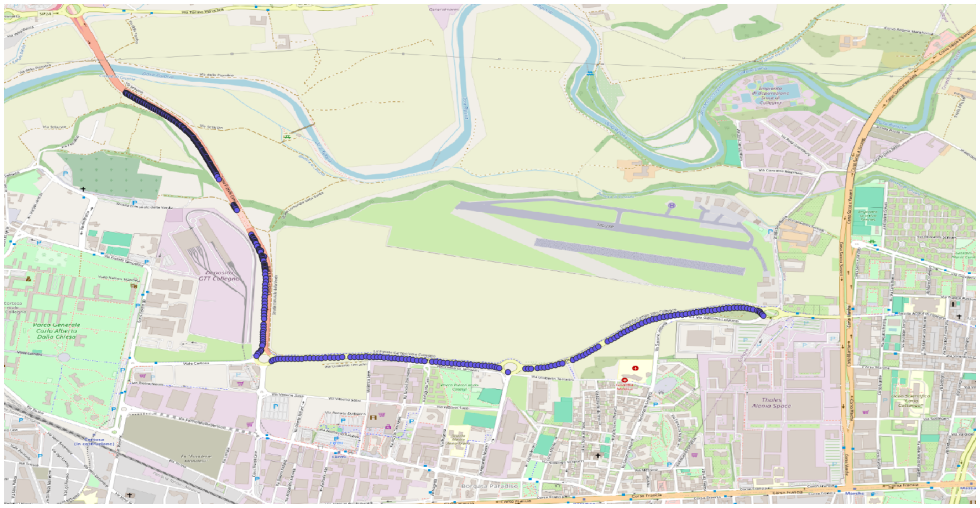


Figure 4.24: Visualization of the guardrails on the official cadastral road map of the Piedmont region. Each point corresponds to a detected guardrail.

By overlaying the guardrail point layer onto the cadastral basemap, it is possible to visualize the exact position of every detected guardrail along the surveyed highway segment. Figure 4.24 shows an example of this overlay, where each point corresponds to a guardrail ROI extracted from the video sequence.

4.7.4 Attribute and Spatial Queries

The GeoPackage database can be queried directly within QGIS to isolate specific subsets of guardrails based on their structural condition or typology. This functionality enables a rapid inspection of critical elements along the route.
For example, by applying an attribute filter on the field `condition`, it is possible to select only the guardrails classified as `Deformed` and visualize their spatial distribution

(Figure 4.25). Similarly, filtering by the field `type` allows the extraction of all `triple-wave` guardrails, as shown in Figure 4.26.

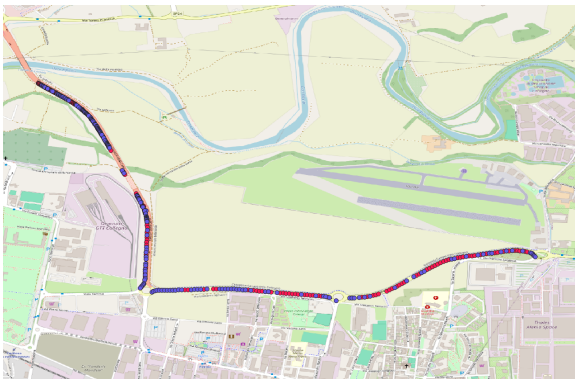


Figure 4.25: Example of attribute query selecting only guardrails classified as `Deformed` - in red.

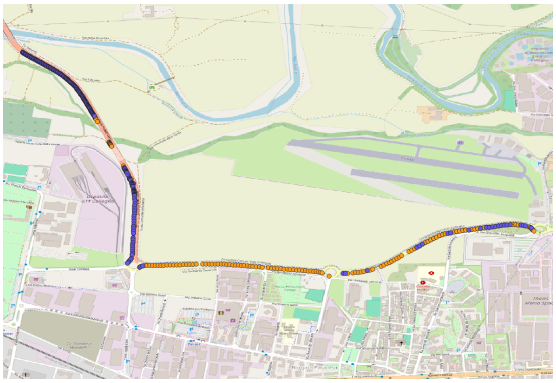


Figure 4.26: Example of attribute query selecting guardrails of type `triple-wave` - in orange.

4.7.5 Linking Guardrail Records to the corresponding frames

To facilitate rapid inspection of the visual evidence associated with each detected guardrail, the spatial database was enriched with a dedicated field containing a direct hyperlink to the corresponding video frame. This functionality enables the operator to open the original image of a guardrail instance directly from the attribute table in QGIS, thus allowing an immediate verification of the model predictions and facilitating manual validation during the case study.

A new attribute, named `frame_link`, was added to the GeoPackage using the QGIS Field Calculator and its widget type was set to *Hyperlink*, enabling clickable links inside the attribute table. When a user selects a point on the map and opens its attribute form, the hyperlink provides direct access to the corresponding frame stored in the project directory. An example is shown in Figure 4.27.

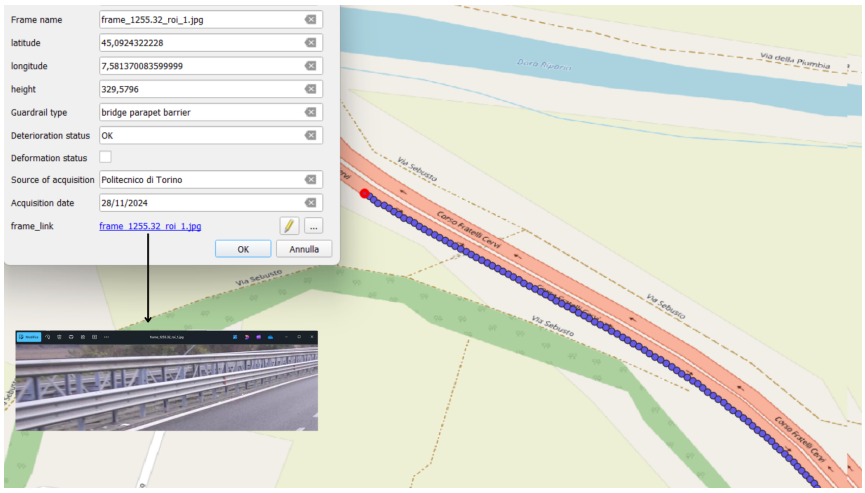


Figure 4.27: Example of a video frame opened through the hyperlink stored in the `frame_link` attribute of the GeoPackage database.

Chapter 5

Google Street View integration

To further assess the robustness and generalizability of the proposed methodology, an **additional validation step** was conducted using external imagery sourced from ***Google Street View*** **(GSV)**. The goal of this experiment is to determine whether the entire processing pipeline—segmentation, classification, and degradation assessment—can be reliably applied to frames that were not captured by the on-board camera system used in the case study. In other words, the same 2.3 km highway segment analysed in Chapter 4 was re-evaluated using a small set of Street View images, in order to verify whether consistent outcomes can be obtained despite the change in acquisition platform, viewpoint, and radiometric characteristics.

5.1 Dataset Acquisition from Google Street View

A total of 15 frames were manually extracted from Google Street View. All images correspond to **June 2025**, just six months after our on-site survey, ensuring that the physical state of the infrastructure remained substantially unchanged between the two datasets. Each image was selected to cover a representative portion of the infrastructure, ensuring that the corresponding guardrails were visible and comparable with those present in the original dataset. Figure 5.1 illustrates the approximate locations along the highway from which the Street View frames were extracted.

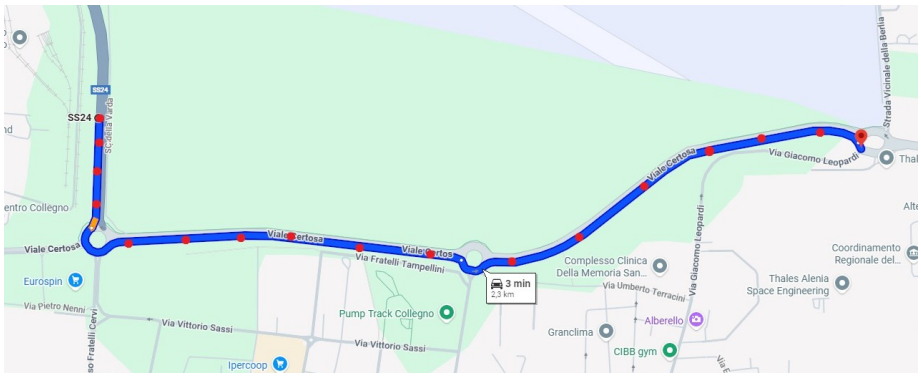


Figure 5.1: Geographical distribution of the fifteen Google Street View acquisition points along the analysed 2.3 km highway segment in Turin. The points are visualised on top of the official road network map provided by the Geoportale Piemonte.

These collected frames were then processed through the **complete pipeline without any modification** or additional fine-tuning: the U-Net segmentation model, the guardrail type classifier, and the condition assessment modules were applied exactly as in the case study.

5.2 Results

The **U-Net segmentation** model demonstrated strong generalization capabilities when applied to GSV images. As shown in Figure 5.2, the guardrail regions were correctly identified in the great majority of cases, with minimal loss of completeness. The most notable difference compared to the on-board dataset lies in the presence of larger radiometric variations and slight blurring in the GSV imagery, which occasionally reduced the sharpness of the mask boundaries. The morphological post-processing step proved effective in correcting small artefacts, leading to clean and contiguous masks that could be reliably used for ROI extraction.

Figure 5.2: Example of segmentation performance on Google Street View frames. From left to right: original image, raw U-Net binary mask, and processed mask after morphological filtering.

The **guardrail classifier** produced highly consistent predictions across the majority of the GSV ROIs (see Figure 5.3). All instances of triple-wave, double-wave, bridge parapet barrier, and central median barrier were correctly identified when the ROI retained sufficient contextual structure.

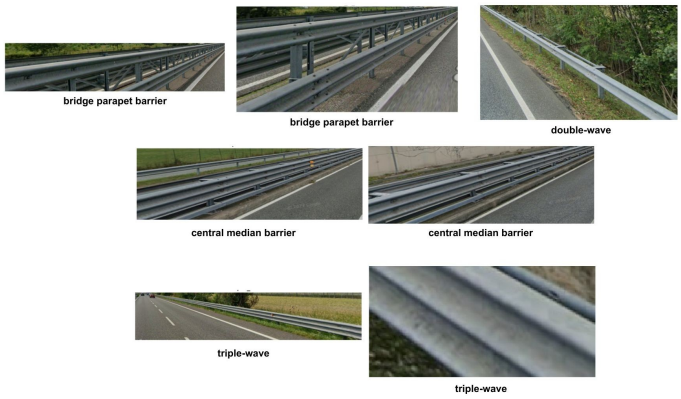


Figure 5.3: Example of correctly classified guardrails.

The primary source of misclassification is illustrated in Figure 5.4. In some situations, the segmentation process extracts only the upper portion of a central median barrier, resulting in an incomplete ROI that lacks the lower structural elements typically distinguishing this category from a triple-wave barrier. When this occurs, the classifier tends to assign the ROI to the latter class, as the visible features closely resemble the learned appearance of a triple-wave profile.

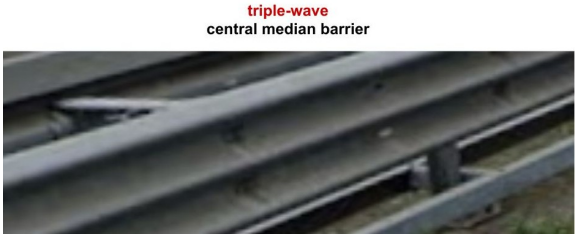


Figure 5.4: Example of the most frequent classification error on Google Street View data: the lower portion of a central median barrier is partially occluded or removed by the segmentation, leading the classifier to assign the ROI to the `triple-wave` class.

These errors highlight an intrinsic limitation of using Street View imagery, where perspective distortions and variable camera heights often reduce the visibility of the lower guardrail components.

The **deterioration analysis module** also generalised well to the GSV dataset. The majority of GSV guardrails were classified as *Ok* or *To be monitored*, with both the visual inspection of the images and the typical condition of guardrails along this road segment.

The **geometric deformation assessment** produced consistent results. The profiles extracted from the segmented guardrails remained largely linear, with deformation values well below the threshold used to flag structural anomalies. This behaviour is expected, given that the analysed segment of the highway does not contain visibly deformed barriers, and confirms that the polynomial fitting method maintains its reliability on external data sources. Minor fluctuations in the estimated deviation are attributable to the oblique viewing angle of GSV cameras and to perspective shortening, both of which increase the sensitivity of the fitted model to small segmentation imperfections.

5.3 Discussion

Although the number of Street View frames is limited and does not allow for a one-to-one comparison with the full on-board sequence, the experiment demonstrates that the proposed methodology can be transferred to external image sources without significant performance degradation. The segmentation remains stable, the classifier

correctly identifies nearly all barrier types, and both deterioration and deformation assessments yield values in agreement with what would be expected from visual inspection. The few observed misclassifications are directly linked to partial visibility of the ROI—probably a limitation inherent to the Street View acquisition process rather than to the models themselves. These results support the feasibility of integrating Google Street View imagery into large-scale or preliminary guardrail monitoring applications, particularly in scenarios where direct on-site video collection is not available.

5.4 Motivation and Potential Applications

The rationale behind this experiment lies in the potential future integration of public geographic data sources into automated infrastructure monitoring systems. If comparable results can be achieved using Street View imagery, the proposed methodology could enable large-scale, low-cost guardrail assessment in situations where direct video acquisition is not feasible. Furthermore, access to Google Street View data through public APIs—when permitted—may open the possibility of performing periodical analyses, even retrospectively, by leveraging the historical imagery stored by the platform.

Chapter 6

Conclusion and Future Works

6.1 Summary

This thesis aims to address a series of open issues:

1. the lack of automated and scalable methods for guardrail detection and assessment;
2. the absence of unified pipelines capable of classifying both guardrail typology and structural condition;
3. the difficulty of integrating visual results with geographic information for long-term monitoring;
4. the need for low-cost and replicable inspection systems that can support public administrations.

To respond to these challenges, a complete pipeline was developed for the automatic detection, segmentation, classification, and georeferencing of roadside guardrails, using low-cost onboard sensors and open data. The system proved effective in extracting guardrail regions from stereo-camera images, evaluating their apparent condition, and associating each detected element with precise GNSS coordinates. Furthermore, the integration of Google Street View images demonstrated that the proposed methodology can be extended beyond proprietary datasets, offering an additional low-cost avenue for visual verification and historical analysis. Overall,

the results confirm that an automated, AI-based approach can substantially reduce the dependence on manual surveys, while improving objectivity and scalability in guardrail monitoring.

6.2 Discussion and Considerations

A central consideration of this work is the idea that AI should be applied where it provides concrete operational value. In the context of road safety, guardrail inspection is both labour-intensive and crucial for public security. Automating even part of this process can therefore deliver tangible benefits.

6.2.1 Strengths and Contributions

The framework directly addresses the critical issues highlighted in the initial problem formulation:

- It offers a guardrail-specific detection pipeline, overcoming the limitations of generic object detection solutions.
- It relies on low-cost and easily accessible sensors, reducing operational expenses and increasing accessibility for public administrations.
- It introduces spatio-temporal synchronisation between the visual stream and GNSS data, filling one of the major gaps identified in existing literature.
- It produces a georeferenced digital inventory, a resource currently lacking in most metropolitan areas, including Turin.
- It demonstrates the feasibility of integrating Google Street View as an additional source for monitoring and validation.

These aspects represent direct and measurable contributions to the research landscape on automated infrastructure inspection.

6.2.2 Limitations

Despite its promising results, the pipeline presents several limitations:

- The reliability of the analysis strongly depends on the accuracy of the segmentation masks. Occlusions (vehicles, vegetation), shadows, and reflections may reduce segmentation quality.
- Deformation assessment is performed in the image domain: apparent geometric irregularities may not always correspond to true physical deformations.
- Variability in perspective, lighting conditions, and road geometry introduces noise that may affect both classification and deformation estimation.
- Real-world conditions along urban roads and highways are extremely heterogeneous, and some scenarios remain challenging for the neural network.

However, by combining multiple frames and exploring multi-source imagery, the pipeline can detect irregularities with a sufficient degree of consistency for operational monitoring.

6.2.3 General Considerations

In a period in which AI systems raise concerns about sustainability and computational cost, this work supports the idea of selective and efficient use of artificial intelligence. Instead of exhaustively processing entire video sequences, the pipeline can act as a filter, flagging only potentially problematic segments for subsequent human inspection. This strategy reduces unnecessary computation and supports more sustainable maintenance workflows.

6.3 Future Works

Several directions may extend the contributions of this thesis:

- **Robust Occlusion Handling:**Future segmentation models may incorporate dedicated modules for occlusion reasoning, temporal consistency, or multi-view fusion, thereby improving robustness in complex scenarios.

- **Fully Automated Integration with Open Data Platforms:** Developing a pipeline that automatically integrates municipal GIS layers, road imagery APIs, and newly acquired video streams would enable continuous monitoring of urban and highway networks.
- **Historical Evolution and Predictive Maintenance:** Leveraging the historical archives of Google Street View would allow longitudinal analyses of guardrail condition, supporting predictive maintenance strategies and asset lifecycle management.
- **Extension to Other Roadside Elements:** The modular nature of the proposed pipeline makes it adaptable to other roadside assets (traffic signs, crash cushions, light poles, drainage systems), paving the way toward a comprehensive digital inventory.
- **Energy-Efficient AI and Edge Computing:** Lightweight convolutional models, deployment on edge devices, and selective model activation could reduce energy consumption and support real-time processing directly onboard vehicles.

Bibliography

[1] National Academies of Sciences, Engineering, and Medicine. *Highway Safety Barriers: Design and Maintenance Guidelines*. Transportation Research Board, 2023. URL: https://nap.nationalacademies.org/read/26763/chapter/3.

[2] Federal Highway Administration. *Automated Guardrail Inventory and Condition Evaluation*. U.S. Department of Transportation, 2022. URL: https://rosap.ntl.bts.gov/view/dot/63368/dot_63368_DS1.pdf.

[3] G. Fabbri, L. Mazzini and A. Martini. "Indexing the Maintenance Priority of Road Safety Barriers". In: *Infrastructures* 8.181 (2023). URL: https://cris.unibo.it/retrieve/9f89d01f-35eb-403b-9c5b-c5593883c3dd/infrastructures-08-00181-with-cover.pdf.

[4] P. Nguyen, T. Miller and D. Song. "Enhancing Road Safety on US Highways: Leveraging Advanced Computer Vision for Automated Guardrail Damage Detection and Evaluation". In: *Buildings* 15.5 (2025), p. 668. DOI: 10.3390/buildings15050668.

[5] *Decreto Ministeriale numero 223 del 18/02/1992*. Local file: file_0000000051ec71f4ba18f5b03ff99f89. Accessed: 2025-11-22.

[6] Strade & Autostrade. *Il catasto funzionale delle barriere di sicurezza*. Accessed: 2025-11-22. 2022. URL: https://www.stradeeautostrade.it/segnaletica-e-sicurezza/il-catasto-funzionale-delle-barriere-di-sicurezza/.

[7] *Quaderno Tecnico ANAS - Barriere di Sicurezza*. Local file: file_00000000e6e07243997e33248f4e031a Accessed: 2025-11-22.

[8] *Decreto Ministeriale numero 2367 del 21/06/2004*. Local file: file_0000000087707243ab6517622e8901 Accessed: 2025-11-22.

[9] Alberto Broggi et al. “Shape-Based Pedestrian Detection”. In: *IEEE Intelligent Vehicles Symposium*. 2005, pp. 215–220. DOI: 10.1109/IVS.2005.1505119.

[10] Alberto Broggi et al. “Multi-Resolution Vision for Automatic Detection of Guard Rails”. In: *IEEE Intelligent Vehicles Symposium*. 2005, pp. 334–339. DOI: 10.1109/IVS.2005.1505112.

[11] Li Fang, Hsing Lee and Hwan Cho. “Guardrail Detection Using Structure from Motion”. In: *Journal of Transportation Engineering* 136.11 (2010), pp. 965–974. DOI: 10.1061/(ASCE)TE.1943-5436.0000164.

[12] David Hernández and Beatriz Marcotegui. “Automatic Detection of Road Safety Barriers from Mobile Mapping LiDAR Data”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 114 (2016), pp. 146–158. DOI: 10.1016/j.isprsjprs.2016.02.011.

[13] Junho Seo, Yichang Tsai and Hui Chia. “Vision-Based Guardrail Condition Assessment”. In: *Transportation Research Record* 2432 (2014), pp. 41–50. DOI: 10.3141/2432-06.

[14] Xiaoqiang Jin, Wei Zhang and Peng Liu. “Improved U-Net for Automatic Guardrail Segmentation”. In: *IEEE Transactions on Intelligent Transportation Systems* 25.6 (2024), pp. 12345–12356. DOI: 10.1109/TITS.2024.3356789.

[15] Qile Zhu, Wanxuan Zhao and Lulin Chen. “Domain Shift in Road Scene Segmentation: A Review”. In: *IEEE Access* 9 (2021), pp. 103915–103930. DOI: 10.1109/ACCESS.2021.3098328.

[16] Xiaoqiang Jin and Wei Zhang. “Guardrail Damage Detection via Lightweight Deep Learning”. In: *Sensors* 24.3 (2024), p. 1023. DOI: 10.3390/s24031023.

[17] Ying Lin, Xinyu Li and Bo Yang. “Guardrail Damage Detection Using Deep Learning and Edge Extraction”. In: *International Conference on Image Processing*. 2021, pp. 3104–3108. DOI: 10.1109/ICIP42928.2021.9506761.

[18] Sagar Goyal and Harshit Kumar. “Automatic Defect Detection in Highway Guardrails Using CNNs”. In: *Automation in Construction* 140 (2022), p. 104331. DOI: 10.1016/j.autcon.2022.104331.

[19] Jian Wang and Xiaodong Li. "Vision–Radar Fusion for Static Infrastructure Detection". In: *IEEE Transactions on Vehicular Technology* 68.2 (2019), pp. 1153–1163. DOI: 10.1109/TVT.2018.2889879.

[20] Yifan Sun and Hui Xu. "Guardrail Detection Using Multi-Sensor Fusion for ADAS". In: *IEEE Sensors Journal* 20.10 (2020), pp. 5634–5645. DOI: 10.1109/JSEN.2020.2969332.

[21] Qian Zhang and Xuyang Huang. "Stereo Vision-Based Highway Asset Detection". In: *Remote Sensing* 12.4 (2020), p. 673. DOI: 10.3390/rs12040673.

[22] Seung Kim and Jihoon Park. "3D Point Cloud-Based Guardrail Recognition". In: *ISPRS International Journal of Geo-Information* 10.8 (2021), p. 542. DOI: 10.3390/ijgi10080542.

[23] *Geoportale Piemonte*. https://www.geoportale.piemonte.it. Accessed via Metropolitan City of Turin GIS Services. 2024.

[24] Autovie Venete. *MOMAS: Mobile Mapping System for Highway Asset Management*. Technical report. 2019. URL: https://www.autovie.it.

[25] Autovie Venete. *Road Management System for Image-Trajectory Synchronization*. Internal software documentation. 2019. URL: https://www.autovie.it.

[26] *Decreto Ministeriale 18/02/1992 n.223 - Norme funzionali e geometriche per la costruzione delle strade*. Ministero dei Lavori Pubblici, Italia, 1992. URL: https://www.mit.gov.it.

[27] F. Serianni. *Le barriere stradali di sicurezza*. Edizioni Scientifiche Italiane, 1993. URL: https://www.edizioniscientificheitaliane.it.

[28] *Istruzioni CNR sulle barriere di sicurezza stradale*. Consiglio Nazionale delle Ricerche, 1992. URL: https://www.cnr.it.

[29] *Decreto Ministeriale 21/06/2004 n. 2367 – Adozione della norma EN 1317*. Ministero delle Infrastrutture e dei Trasporti, 2004. URL: https://www.mit.gov.it.

[30] *EN 1317-2: Road restraint systems — Performance classes, impact test acceptance criteria and test methods for safety barriers*. CEN - European Committee for Standardization, 2010. URL: https://standards.cencenelec.eu.

[31] *Manuale Tecnico delle Barriere di Sicurezza*. ANAS S.p.A., 2019. URL: https://www.stradeanas.it.

[32] *Regulation (EU) No 305/2011 – Construction Products Regulation (CPR)*. European Parliament and Council, 2011. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32011R0305.

[33] Hayes E. Ross et al. *Roadside Design Guide*. 4th ed. AASHTO, 2008. URL: https://store.transportation.org.

[34] Simone Franceschini et al. "Automated visual inspection of roadside safety barriers". In: *Transportation Research Record* 2674 (2020), pp. 450–462. DOI: 10.1177/0361198120918417. URL: https://journals.sagepub.com/doi/10.1177/0361198120918417.