Deliverable D 2.2
WP2 Report on AI approaches and models

Project acronym:	RAILS
Starting date:	01/12/2019
Duration (in months):	36
Call (part) identifier:	H2020-S2R-OC-IPX-01-2019
Grant agreement no:	881782
Due date of deliverable:	Month 31
Actual submission date:	June 30th 2022
Responsible/Author:	Stefania Santini (CINI)
Dissemination level:	Public
Status:	Issued

Reviewed: no

Document history		
Revision	*Date*	*Description*
1	June 20^{th} 2022	First issue for internal and AB review
2	June 29^{th} 2022	Second issue for EU review

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Stefania Santini	CINI	WP2 Leader
Lorenzo De Donato	CINI	Contributor
Valeria Vittorini	CINI	Contributor
Francesco Flammini	LNU	Contributor
Rob Goverde	TUDELFT	Internal Review

Advisory Board Reviewers	
Name	*Company or Institution*
Milan Banić	University of Niš, Republic of Serbia, and S2R SMART2 project
Stefano Delucchi	Aitek S.p.A.

Funding

This project has received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement n. 881782 Rails. The JU receives support from the European Union's Horizon 2020 research and innovation program and the Shift2Rail JU members other than the Union.

Disclaimer

The information and views set out in this document are those of the author(s) and do not necessarily reflect the official opinion of Shift2Rail Joint Undertaking. The JU does not guarantee the accuracy of the data included in this document. Neither the JU nor any person acting on the JU's behalf may be held responsible for the use which may be made of the information contained therein.

Contents

Executive Summary

This Deliverable leverages the results achieved in Deliverable D2.1 to address AI approaches customised to the rail sector. On the basis of the pilot cases studies selected in Deliverable D2.1, i.e., railway obstacle detection and collision avoidance, and cooperative driving for virtual coupling of autonomous trains, specific proofs-of-concept are provided to investigate the adoption of AI methods for automatic and safe train operations.

Specifically, the report addresses the following themes for each case study: i) a description of the background to identify the main objectives and the open issues; ii) the identification of proper research questions; iii) the selection of safety and performance indicators to be used as evaluation criteria; iv) the analysis of technology and methodological alternatives; v) a high-level description of the selected AI approach and its possible adaptation to railway; vi) the identification of the main technological and architectural issues to be considered by future implementation.

The proposed proofs-of-concept are intended to give a first insight towards the definition of qualitative and technology roadmaps which could lead to the deployment of AI applications aiming at enhancing rail safety and automation.

Abbreviations and acronyms

Abbreviations / Acronyms	Description
ACC	Adaptive Cruise Control
AD	Anomaly Detection
AI	Artificial Intelligence
ATC	Automatic Train Control
ATO	Automatic Train Operation
ATP	Automatic Train Protection
DDPG	Deep Deterministic Policy Gradient
DE	Distance Estimation
DL	Deep Learning
DMPC	Distributed Model Predictive Control
DNN	Deep Neural Networks
DPG	Deep Policy Gradient
DQN	Deep Q-learning Network
DRL	Deep Reinforcement Learning
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
FPS	Frame Per Second
GAN	Generative Adversarial Network
GPU	Graphic Processing Unit
GTA V	Grand Theft Auto V
ITC	Intelligent Train Control
ITO	Intelligent Train Operation
ITP	Intelligent Train Protection
JU	Joint Undertaking
KPI	Key Performance Indicator
Maas	Mobility-as-a-Service
MARL	Multi-Agent (Deep) Reinforcement Learning
ML	Machine Learning
MPC	Model Predictive Contro
OD	Object Detection
R-CNN	Region-based CNN
RD	Rails Detection
RGB	Red, Green, Blue
RoI	Region of Interest
RQ	Research Question
RBC	Radio Block Center
RL	Reinforcement Learning
T2T	Train-to-Train
VC	Virtual Coupling
VCTS	Virtually Coupled Train Set

1. Background

The present document constitutes the Deliverable D2.2 “WP2 Report on AI approaches and models” of the Shift2Rail JU project “Roadmaps for AI integration in the Rail Sector” (RAILS). The project is in the framework of Shift2Rail’s Innovation Programme IPX. As such, RAILS does not focus on a specific domain, nor does it directly contribute to specific Technical Demonstrators but contributes to Disruptive Innovation and Exploratory Research in the field of Artificial Intelligence within the Shift2Rail Innovation Programme. The successor of the Shift2Rail Joint Undertaking is currently the Europe’s Rail Joint Undertaking (EU-Rail) established by Council Regulation (EU) 2021/2085 of 19 November 2021.

The RAILS workpackage WP2 investigates the adoption of learning techniques and other AI methods for **enhanced rail** safety and automation. The present Deliverable describes the work carried out in task 2.2 (Development of AI approaches and models) whose aim is to develop AI approaches according to the results and choices reported in D2.1:

- Directions for transferability, including a) Intelligent Train Operation, and b) AI-aided Cooperative Driving;
- Pilot case studies, specifically: Obstacle Detection and Collision Avoidance, and Cooperative Driving for Virtual Coupling of Autonomous Trains.

The two pilot case studies aim at providing the context for proofs-of-concept in order to investigate the advantages and benefits that AI could provide to rail safety and automation and the gaps to be fulfilled. In the development of the proofs-of-concept, transferability of concepts and techniques from other sectors are considered, in particular from the automotive sector and unmanned ground vehicles.

The above-mentioned transferability directions are part of a vision addressing the *safe* applicability of AI techniques to train control systems, starting from traditional Automatic Train Control (ATC) concepts and moving towards fully autonomous train driving in open environments.

Such vision has been introduced in D2.1 and further described in [1]. Fig.1.1 provides an overview of the Intelligent Train Control (ITC) concept, which integrates Intelligent Train Operation (ITO), ATP, and – possibly – Intelligent Train Protection (ITP).

If used in conjunction with Automatic Train Protection (ATP), ITO can safely extend the functionalities of traditional ATO to account for intelligent and adaptive behaviours, in order to optimise passenger comfort, energy consumption, and line capacity, e.g., through Virtual Coupling [2]. In fact, just as ATO, ITO actions would be supervised by the ATP following the safety envelope architectural pattern [3]: for example, in case ITO is intended to take an action that would bring the system into a unsafe state, the ATP will block that action and bring the system into a fail-safe state. This would also allow certification against reference safety standards using existing traditional approaches. However, this would be possible as long as the ATP is available and perfectly functioning.

In case ATP is not available, such as in many secondary railway lines or in developing countries, or when ATP failures oblige to partial supervision, it would be possible to exploit AI as to improve situation awareness. For example, ITP could act as an advanced driving assistance system (as happens within automotive [4]) by implementing functionalities such as

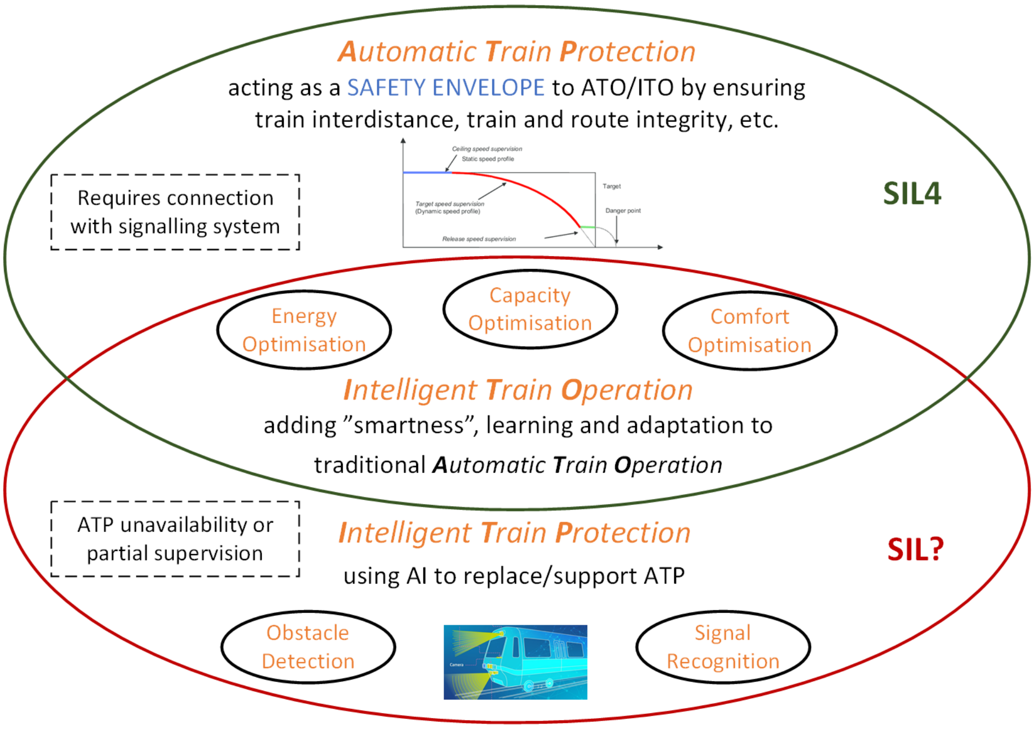


Fig. 1.1. Intelligent Train Control (ITC)

signal recognition and obstacle detection. Otherwise, even if ATP is available and perfectly working, ITP can be a useful complement to detect events that are not managed by ATP (e.g., obstacles). Worth highlighting, functionalities like obstacle detection and signal recognition, possible implemented through artificial vision, radar, LiDARs and other appropriate sensors, can also aid special manoeuvres such as those needed to enable Virtual Coupling, by implementing what is known as "adaptive cruise control" in automotive (e.g., [5]).

Despite the fact that the implementation of ITP systems would represent an important step towards *fully autonomous train driving*, significant certification concerns arise, as ITP is expected to directly involve AI in safety-critical applications. It is well known that the certification of these intelligent systems against international standards is very challenging, even if efforts are being carried out towards the definition of ad-hoc standards and regulations for AI in railways, taking also into account eXplainable AI approaches as one of the main enablers that could help to improve transparency and trustworthiness of intelligent systems [6].

Both the pilot case studies proposed in RAILS WP2 are conceived as benchmarks in this vision of Intelligent Train Control.

2. Objective

The objective of the RAILS workpackages WP2, WP3, and WP4 is to define pilot case studies and develop proofs-of-concept leading to a technology roadmapping for an effective pick-up of AI in the rail sector. Specifically, the project activities reported in this document and in the next Deliverable D2.3 address the Objective 4 and 5 of the RAILS project:

- *Development of methodological and experimental proof-of-concepts*;
- *Development of Benchmarks, Models and Simulations*.

This deliverable relies on the state-of-the-art conducted in WP1, taking direct input from Deliverable D2.1 (“Report on case studies and analysis of transferability from other sectors”) and formalising the pilot case studies identified in D2.1, so providing feasibility studies for the adoption of AI and related techniques in the area of rail safety and automation.

The case studies discussed in this deliverable deal with:

- obstacle detection;
- cooperative driving.

They are very different from each other: obstacle detection is currently being studied in a number of application domains, including the rail sector, whereas cooperative driving in railways tackles with very innovative and challenging scenarios, such as Virtual Coupling. Hence, they allow to investigate the potentiality of AI in railways from very different points of view. Indeed, the ultimate goal of the proofs-of-concept is to help identifying gaps and opportunities, they are not expected to provide solutions to a specific problem. On the other hand, both the case studies are intended to be the context for acquiring the necessary knowledge to go a step further the current state of the play, towards a vision of safe automation and intelligent train control.

Therefore, as for the first case study, most of the solutions dealing with the usage of AI for obstacle detection in railways are briefly recalled in the deliverable, and the specific objective is to delineate the framework in which the experimentation will be carried out. The proposed approach also introduces some elements of novelty with respect to the state-of-the-art, but its main objective is to provide a benchmark for AI techniques to the purpose of environment perception and situation awareness, also leveraging on the transferability of applications usually adopted for different purposes, such as anomaly detection.

As for the second case study, there is not previous work to be considered in proposing AI approaches to cooperative driving for Virtual Coupling. The objective is to explore a new concept of railway transport basing on the work done for car-following scenarios, in order to identify promising research directions in cooperative driving for autonomous systems.

3. Introduction

Deliverable D2.2 provides a detailed description of the AI solutions and approaches for addressing the problems and the challenges posed by the case studies, the related models and metrics, the technological and operational issues. Hence, this Deliverable reports on the technological and methodological possible solutions, alternatives and criticalities to be addressed by subsequent implementations.

The heart of this document consists of two main chapters, addressing the above mentioned issues for the two pilot case studies: Chapter 4 is devoted to Obstacle Detection for Collision Avoidance and Chapter 5 deals with Cooperative Driving for Virtual Coupling.

The two chapters share the same structure: a brief presentation of the scope of the selected case study is provided in Sections 4.2 and 5.2; the specific objectives of the related proof-of-concept and the research questions to be addressed are introduced in Sections 4.3 and 5.3; a description of the proposed approach and tools is given in Sections 4.4 and 5.4, as well as a presentation of the AI techniques being exploited in the development of the proof-of-concept in Sections 4.6 and 5.5. How to cope with the need of data is discussed in Sections 4.5 and 5.6 respectively. Finally, expected results and possible criticalities are discussed in Sections 4.7 and 5.7.

The implementation and experimentation activities are ongoing according to the objectives, approaches and techniques reported in this document. Details and results of these activities are the object of the next Deliverable D2.3 ("Report on experimentation, analysis and discussion of results").

4. Railway Obstacle Detection and Collision Avoidance

4.1. Introduction

Perceiving the environment is one of the fundamental requisites to move towards full autonomous train driving, especially in open railway environments (i.e., railways that are not completely isolated from external factors) [1]. In order to move towards fully autonomy, trains should be equipped with adequate sensors and systems that, besides being compliant with standards and certifications, make trains capable of achieving **full situation awareness** in relation to the health status of the internal components of the vehicle and external threats or signals. In this chapter, we focus on defining the challenges that should be overcome, highlighting some requirements that must be taken into account, and identifying AI approaches that could help to cope with the obstacle detection and collision avoidance task in the context of what we defined as Intelligent Train Control [6].

4.2. Background and Description

Obstacles, or objects in general, may assume different meanings in different domains (e.g., railway, automotive, and avionics). For example, for unmanned aerial vehicles, obstacles like buildings contribute to the identification of the possible paths; also, in automotive, cars parked at the side of the lane contribute to the delineation of the lane that can be travelled by running vehicles. In railway, as the path of the train is inflexibly defined by tracks, obstacles on the railroad only represent threats.Furthermore, the concept of collision avoidance also changes. Differently form cars or aerial vehicles which could avoid obstacles by steering, trains cannot literally "avoid" obstacles, they can only avoid the impact by braking. The stopping distance plays a central role in this context. A train requires to detect the obstacle at hundreds of metres (or even kilometres) of distance to avoid the impact (or mitigate consequences) as it needs hundreds of meters to full stop even at reduced speed (70/80 km/h) [7, 8]. Table 4.1 summarises the differences between railway, automotive, and avionics according to four of the main characteristics that should be taken into account when it comes to "avoid" collisions: i) possible actions that can be taken to avoid obstacles, ii) determinism of the path, iii) threat occurrences, and iv) required stopping distance.

Table 4.1: Domains' characteristics related to collision avoidance.

Domain	Railway	Automotive	Avionics
Possible Actions	1 - brake.	3 - left, right, brake.	5^+ - top, down, left, right, decrease speed, (+ oblique directions).
Path	Inflexible, limited by rail tracks.	Flexible within the carriageway/lane.	Not limited if not by obstacles.
Threat Occurrence	Uncommon/rare at connection points (level crossings, stations). Very rare elsewhere.	Extremely common in urban environments. Common on highways (other vehicles are also obstacles).	Almost null in open aerial space. Common in urban space / at low heights.
Stopping Distance	Hundreds of meters / kilometres.	Typically, in the order of few hundreds meters (100-200m).	Aerial vehicles do not properly brake.

Therefore, from what described in Table 4.1, we can asses that the railway domain is significantly different from the others as:

- The determinism of the “inflexible” path limits the possible actions that can be take to avoid obstacles.
- Obstacles on the rail tracks are supposed to be quite rare or, at least, limited to some connection points such as level crossings and train stations.
- The space the train needs to full stop is much longer than that required by a car (given the same speed) given, among others, the following aspects: i) the *low friction of the wheel-rail interface* (if compared to the friction of rubber tires on roads); ii) *passengers' comfort*, sudden braking would make the journey extremely uncomfortable and could also endanger the safety of passengers; and iii) *trains' load*, especially in case of freight trains, as the larger the load the more braking space is required.

In addition to these domain-specific characteristics and constrains, other considerations are required in relation to the AI application that is intended to be used for obstacle detection. Typically, Deep Learning (DL) approaches for obstacle detection [8] rely on architectures trained in supervised mode (i.e., the class of the objects intended to be detected are established and known a-priori). Hence, they are somehow limited as they cannot properly detect objects belonging to classes which have not been considered in the training phase. As we will discuss in the following sections, unsupervised methods oriented at detecting anomalies not known a-priori could be used as complementary approaches to the affirmed supervised ones as to improve robustness and coverage of the whole obstacle detection system.

On the bases of this analysis, we defined in Table 4.2 three main Key Performance Indicators (KPIs) that should be taken into account when evaluating an AI-based obstacle detection system.

Table 4.2: KPIs for the evaluation of AI-based obstacle detection systems.

KPI	Description
KPI1	**Detection Distance.** Indicates the ability of the system to detect distant obstacles in order to mitigate as much as possible the effects of any impact. Measured in meters.
KPI2	**Obstacle Coverage.** Indicates the coverage of the system in terms of the types of obstacles it can detect. Measured in number of objects' classes (limited in case of supervised approaches).
KPI3	**Computation Time.** Indicates the time required by the system to detect possible obstacles on the tracks. Measured in milliseconds.

4.3. Objectives and Research Questions

Through the analysis of this case study and the realisation of the related proof of concept, we mainly aim at defining directions, tools, and AI approaches that could be useful also to solve similar tasks (e.g., intrusion detection at level crossings). Practically, we aim at understanding how it would be possible to overcome (or mitigate) current challenges, fill possible gaps, and outline guidelines and roadmaps as to facilitate the work of future researchers addressing obstacles or intrusions detection tasks in railways.

Despite the fact that also signal recognition is of extremely importance when it comes to improve trains' situation awareness, we decided to *specifically focus on obstacle detection and collision avoidance leveraging data from train ego-perspective cameras* in the context of Intelligent Train Control. Beyond what was already discussed in our previous deliverable [6], the choice of the case study comes also from the fact that, in our view, it would allow us to investigate different AI solutions (from object detection to anomaly detection and distance estimation, as we will discuss in the following sections) and would probably introduce some additional challenges with respect to signal recognition.

On the other hand, the constraint on the camera sensors come from two main factors. First, other Shift2Rail/EU-Rail projects both closed (e.g. GoSAFE RAIL [9], SMART [10]) and ongoing (e.g., SMART 2 [11]) have already faced or are currently addressing the obstacle detection task by means of complex multi-sensor systems. Therefore, not to retrace their path, we will propose a multi-modular framework as to improve robustness and coverage (KPI2) of systems relying on data coming from camera sensors (mainly RGB cameras). Second, camera sensors and AI approaches would allow for more cost-effective solutions; especially when considering other obstacle/intrusion detection tasks in other scenarios (e.g., at level crossings).

To formalise the objectives we intend to achieve with this case study, we have established the following Research Questions (RQs):

RQ1: How can we detect known or unknown obstacles, such as rocks, vehicles, trees and people, in front of the train by using train front cameras and artificial vision?

RQ2: Can obstacle detection solutions based on artificial vision be transferred and adapted from other sectors (e.g., automotive, avionics, robotics, etc.) to railways?

RQ3: Can we demonstrate possible answers to RQ1 and RQ2 through a simple proof-of-concept demonstrator in order to inspire future developments and a technology roadmap?

4.4. Methodology and Tools

As from the comprehensive analysis performed in [8], vision-based obstacle detection systems topically go through two phases: first, rail tracks are detected (to identify the region of interest - RoI); then, objects/obstacles in or near the RoI are recognised.

In our view, this methodology should be extended to improve the systems' performances in relation to KPI2. Most of DL systems are trained in a supervised manner. Architecture families such as YOLO or region-based CNNs (R-CNNs) - commonly known as object detectors - have shown incredible performance when it comes to detecting objects; additionally, they have already been widely investigated for maintenance/inspection applications in the rail sector [12,13]. However, in the training phase, they should be fed with labelled data (objects bounding boxes and labels). This means that if an obstacle, that has not been elaborated in the training phase, appears on the track, it will most likely not been detected. This is an extremely sensitive issue; it would be nearly impossible to take into account (a-priori) every object that could possibly occupy the rail tracks. Common objects could be vehicles, pedestrians, animals, etc; however, also these "classes" are composed of different sub-classes (e.g., cars, motorbikes, and trucks for the vehicles class). Therefore, building a comprehensive dataset that would incorporate all the possible classes and sub-classes is extremely time consuming and difficult as hundreds of kinds of objects should be considered.

To partially overcome this issue, it would be possible to leverage systems oriented at detecting anomalies. Visionary, an anomaly detection system will learn characteristics related to “free” tracks and, in case an uncommon scenario is presented in input, it will produce an anomaly map indicating where the anomaly is located (however, it will not specify the class of the anomaly). As better explained in the following of this manuscript, an anomaly detection system could improve the robustness and the coverage (KPI2) of the whole obstacle detection system as it could:

- support the decision taken by an object detector by confirming its output (i.e., it will highlight an anomaly in the same position as an object detected by the object detector);
- act as a complementary system as it would be capable of detecting unknown anomalies that the object detector would not be able to detect.

For the sake of simplicity, in the following, we will refer to items we would know a-priori as “objects” and to items we would not know a-priori as “anomalies”. Instead, with “obstacles”, we will refer to any kind of item (whether known or unknown).

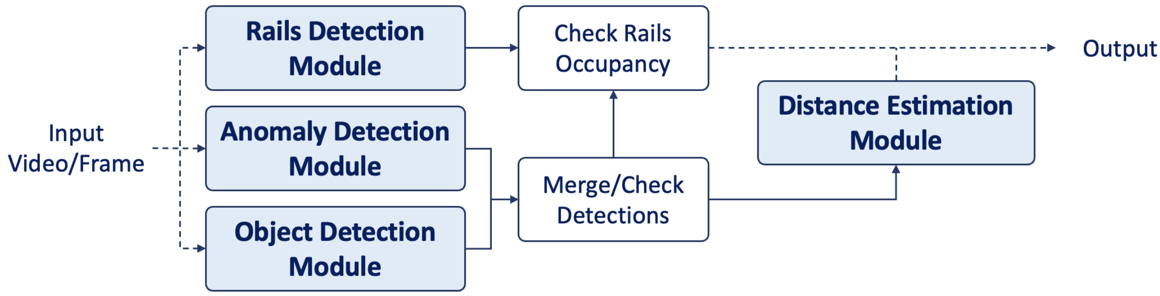
4.4.1. Proposed Methodology

Bearing in mind what was discussed above, in the context of autonomous train driving, an obstacle detection architecture should involve at least four modules:

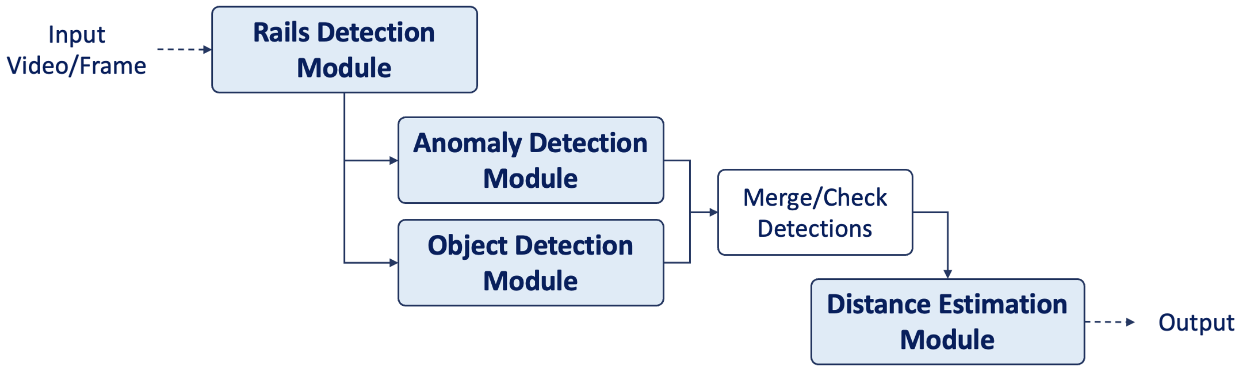
- **Rails Detection (RD) Module**, which is in charge of detecting the rails and define the Region of Interest (RoI).
- **Object Detection (OD) Module**, which aims at identifying, locating, and classifying objects.
- **Anomaly Detection (AD) Module**, which is oriented at detecting and locating anomalies.
- **Distance Estimation (DE) Module**, which estimates the distance between the train and the detected obstacles.

Which is the best configuration to interconnect these systems depends on real-time requirements (KPI3) and computational power availability:

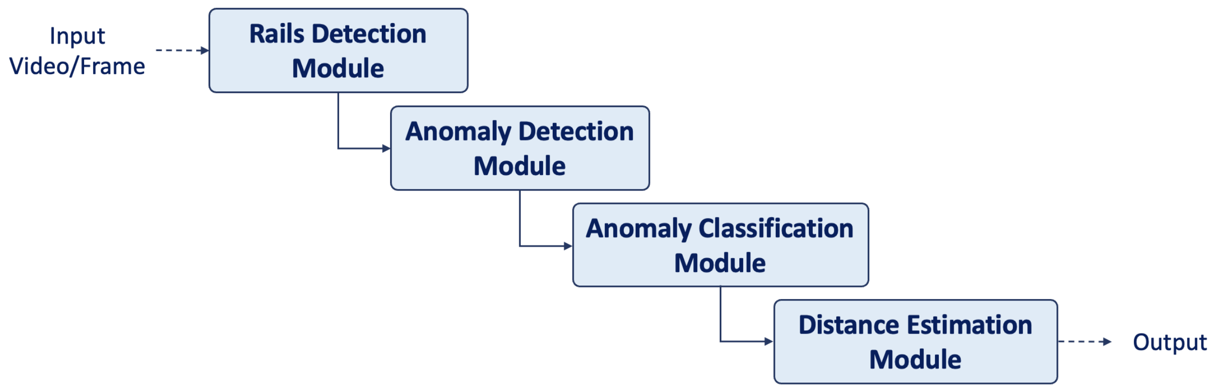
- Conceptually, all the modules could be run in parallel (Figure 4.1(a)); however, in this case, there could be some computational overhead given the fact that the OD and AD modules should look for obstacles within the whole video frame and the fact that all the modules should be run at the same time. Concerning the functioning, briefly, once the RD, OD, and AD modules have produced their outputs, these will be post-processed by the “Merge/Check Detections” module, which will merge the detection from the OD and the AD module, and by the “Check Rails Occupancy”, which will check whether the detected obstacles occupy the tracks or not. Notably, the DE module could be ran in parallel with the “Check Rails Occupancy” module; however, in case the obstacle does not occupy the tracks, there will be an unnecessary computation that *might* delay the subsequent detection.
- Differently, in case of semi-parallel architecture (Figure 4.1(b)), the RD module is run before the OD and AD ones and the obstacles should be detected only within a portion of the original video frame (i.e., the RoI). This would reduce the computation overhead discussed above and would avoid unnecessary computations; however, the overall detection time could sightly increase given the non-parallelism of the RD module.



(a) Parallel Architecture



(b) Semi-parallel Architecture



(c) Sequential Architecture

Fig. 4.1. Obstacle Detection Architectures

- Lastly, the sequential architecture (Figure 4.1(c)) expects all the modules to be run in series; also, it would involve an Anomaly Classification (AC) module instead of the OD module. In brief, when an anomaly is detected by the AD module, it will be cropped out from the frame and classified by the AC module as to recognise (if known) the class it belongs to.

The main problem introduced by the latter configuration is that we would loose the module redundancy introduced within the others. Indeed, the AD module could act as a backup module in case the OD module fails in detecting an object; clearly, the object will not be classified, but it will be at least detected. It may also happen that if the AD module fails in detecting an anomaly, the OD module could still be able to detect the related object, however, this consideration holds if and only if the object is known. Therefore, it would be advisable to adopt parallel modules as to improve the robustness and the coverage (KPI2) of the whole system.

Worth mentioning, for a more advanced architecture, there could be introduced also an Obstacle Tracking Module to detect, in advance, possible obstacles that are moving towards rail tracks. Furthermore, to cover each possible track segment in front of a running train, **cameras mounted in front of it might not be sufficient**, for example, to detect obstacles after a curve. *A more complex multi-systems solution involving on-board, trackside, and even airborne detection systems [14] would be needed*; however, as already mentioned, such a complex architecture is out of the scope of this manuscript.

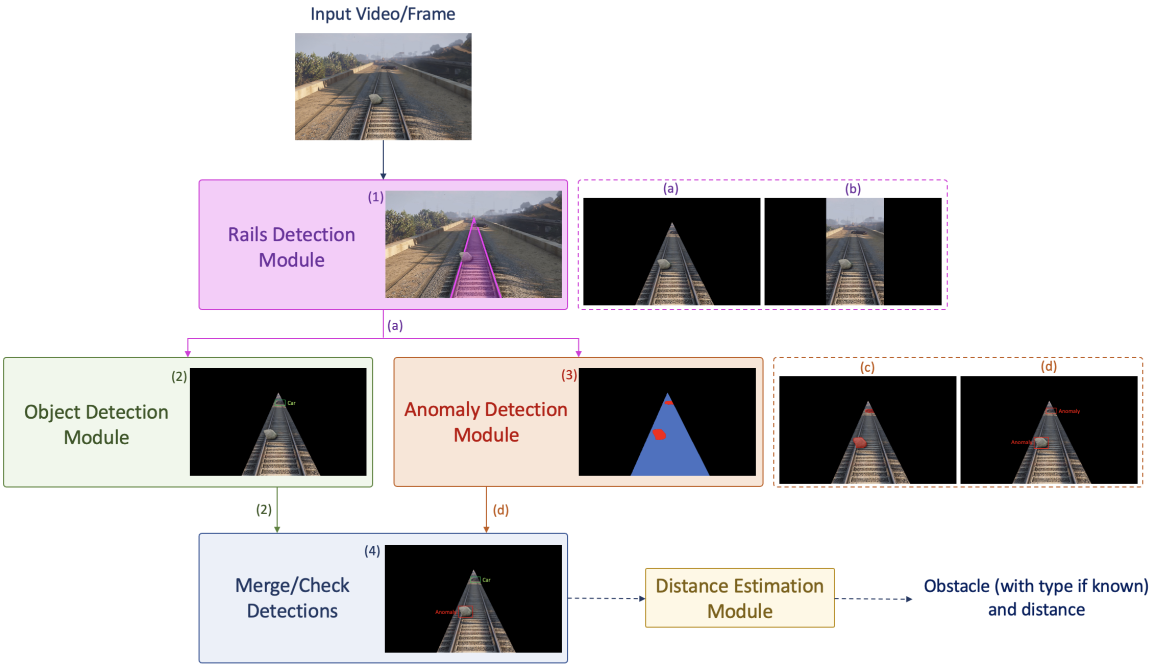


Fig. 4.2. Proposed Framework for Obstacle Detection. The sample image has been taken from Grand Theft Auto V (GTA-V).

As stated above, it is quite difficult to understand which configuration will perform the best in terms of accuracy and real-time performance. These metrics strongly depends on the complexity of the task (e.g., object detection against image classification) and the complexity of the implemented DL approaches. What, in our opinion, could help to make a preliminary decision, is the fact that with a parallel or semi-parallel architecture we would introduce a sort of modular redundancy (as better explained in the following of this manuscript) that could help to improve the robustness of the whole architecture. As for the differences between the parallel and the semi-parallel architecture, the RD, OD, and AD modules could most likely be implemented by exploiting the same algorithms. However, first detecting the RoI and then identifying obstacles could help facilitate the implementation of OD and AD modules, while remaining extremely relevant for the definition of roadmaps and guidelines. Therefore, in this study, we will focus on the semi-parallel architecture, which is better depicted in Figure 4.2 and described below:

- First, the video frame is processed by the RD module that detects the rail track (i.e., the RoI). At this point, post-processing is needed and it is possible to proceed by: cropping out all the background and taking into account a portion of the image which is slightly bigger that the RoI as to also detect potential obstacles standing really close to the rail track (alternative (a) in the hatched purple rectangle), or, by taking into account the smallest vertical portion of the frame which includes the rail track (alternative (b)). For example purposes, in the following, we will take into account alternative (a).

- This image is then processed, in parallel, by the OD and AD modules. The OD module will be able to detect only the objects it has seen in the training phase; in this case, we suppose it has not been trained with “rock” samples, hence, it is capable of detecting - i.e., locating and classifying - only the car (image (2) in the green rectangle). On the other hand, the AD module, which is supposed to be capable of detecting any anomaly, will produce an anomaly map (image (3) in the orange rectangle) highlighting anything different from rail tracks. Also in this case, post-processing is required. It would be possible to segment anomalies (alternative (c) in the hatched orange rectangle) or draw bounding boxes around the anomalies (alternative (d)). In the (d) case, which in our view is the best choice, a class “anomaly” will be associated with each bounding box independently.
- Then, the *Merge/Check Detection* block will take in input (2) and (d) and check if there is any overlapping detection. For the obstacles which have been detected by both the OD and AD modules, it will take into account the detection produced by the OD module (as it also produces the class); otherwise, it will consider the detection produced by the AD module. Note that this block does not involve any AI functionality.
- As the last step, the distance from the obstacle(s) is estimated by the DE module.

Taking into account this framework, in the following of this chapter, we will highlight which, in our view, are the best AI approaches and methodologies to deign each module.

4.4.2. Recommended Tools

In any case, we would need a suitable dataset to produce proofs-of-concept. To that aim, videos recorded on the field (with a camera mounted in front of the train) would help to train and evaluate the aforementioned modules with real data, hence, we would have a more precise indication on how they would work in real environments. However, to efficiently evaluate the performance of ML/DL models, as we will see in the next section, we would also need frames containing obstacles. Therefore, to collect data related to real scenarios, we should introduce real obstacles on the tracks which would be “safely” possible only when test tracks are available (which is not our case). Safer methods would be: i) augmenting real videos with no obstacles by artificially introducing them, or ii) leverage research-oriented simulators or video games.

For research-oriented simulators we mean simulators specifically designed for scientific/testing purposes; an example is RoadRunner[1] for automotive. They would probably be more user friendly than video games when it comes to fully customise the test environment (e.g., placing obstacles in specific positions); on the other hand, video games - such as Grand Theft Auto V (GTA V)[2], Train Simulator[3], and Train Sim World[3] - offer high graphic characteristics but the customisation of the environment is not so straightforward as they have been designed for other purposes (e.g., game-wise train driving simulation). However, worth mentioning, GTA V has been already investigated for self-driving cars [15–17]

[1] https://www.mathworks.com/products/roadrunner.html
[2] https://www.rockstargames.com/it/gta-v
[3] https://dovetailgames.com/products

and may represent a suitable solution to collect data for our task in absence of advanced research-oriented simulators.

4.5. Reference Datasets

In order to properly characterise ML/DL models - i.e., provide the model with all the data necessary as it will be able to efficiently complete the task for which it was designed - datasets for obstacle detection must encompass obstacles placed on the tracks. Many studies in the literature adopt some augmentation techniques to artificially introduce obstacles on free tracks (i.e., without any obstacles) [8]. However, this is a quite tough task as an object should be properly introduced in multiple frames taking also into account the perspective of the scene (see, for example, the GAN-based approach described in [18]). In our view, a dataset that can be exploited together with such an "augmenting" approach is RailSem19 [19] which, for the sake of knowledge, is available[4] together with semantic annotations. Other datasets that can be partially exploited, mainly to pre-train AI models [20], are the well-known ImageNet [21] and Microsoft COCO [22]. Despite the fact that these are not directly tailored to the railway environment, they contain different images and annotations related to different classes of objects (e.g., cars, trains, persons) that can be exploited when dealing with the object detection task. However, as far as we know, no dataset is available in the literature that perfectly matches our task [23]. Therefore, we will most likely exploit simulators or video games to develop this proof-of-concept.

In the following, we report some characteristics that a dataset should have to properly train ML/DL models. First, it would be advisable to fix the track line under examination. Perhaps, when possible, frames coming from multiple rail lines can be used to pre-train the models; however, it would be advisable to fine-tune a model or train it directly from scratch by taking into account data related to the scenario where it will operate.

Then, once the track line has been "fixed", the following specifics should be taken into account:

- **Weather Conditions.** Data should be collected under different weather conditions (e.g., sunny, rainy, cloudy, foggy). Worth mentioning, there could be used some data augmentation approaches, both based on traditional computer vision[5] or GAN (e.g., [24]), to obtain data related to a given weather condition starting from another weather condition; alternatively, other approaches [25] may be exploited to generate new images starting from those collected. However, the main advice here is to avoid introducing data related to extremely rare conditions. For example, if it rarely snows, it would be better not to include snow-related data in the dataset that will be used to train the main model. In this case, it would be advisable to build two different datasets: one without snow-related data (dataset A) and another that encompasses them (dataset B). Hence, the same AI architecture will be trained twice, once on dataset A - and this will be used as the main system - and once on dataset B - which will be used as a complementary system in case of snowy conditions. This will allow the model trained on dataset A not to be "distracted" by relatively superfluous data, and would also increase the robustness of the whole system given the modular redundancy.
- **Light Conditions.** This is related to both day/night light conditions and interference

[4] https://wilddash.cc/railsem19

[5] https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library

conditions (e.g., camera obstructed by sun light). Also in this case, data augmentation approaches can be adopted to obtain, for example, night images starting from day images (e.g., [26]). Furthermore, as already discussed for the weather conditions, it would be possible to build two different instances of the same model trained in light-mode (i.e., with data collected during the day) and in dark-mode (i.e., with data collected during the night or in tunnels). In this way, light features will not interfere with dark features (and vice-versa); however, in this case, it would not be straightforward to manage transitive situations (i.e., sunrise and sunset). Hence, to improve robustness, it would be advisable to rely on multiple kind of cameras (e.g., RGB - perhaps with night vision functionalities -, thermal, etc.). Then, we would have different instances of the same model trained on data coming from different camera sensors to differentiate failure modes and hence improve system robustness. On the other hand, each dataset should include possible interference due to, for example, light reflection.

- **Obstacles.** Assumed that it would be quite impossible - although advisable - to include all the possible obstacles within a dataset, it should at least contain obstacles of different sizes standing in different positions in respect to rail tracks. In case of datasets for object detection, it would be advisable to collect thousands of samples for each class. Otherwise, in case of datasets for anomaly detection (as intended in Section 4.4.1), the class is not particularly relevant. What is relevant, instead, is to diversify the dataset in terms of dimensions and positions of the obstacles to evaluate if the model will perform correctly in most of the cases.

  In this direction, it is possible to identify three main keywords: **distance**, **size**, and **position**.

  The *distance* variation would be somehow intrinsically included when building the dataset (especially if simulators or videogames are used) as, once the obstacle has been placed on the the tracks, it will be automatically recorded at different distances while the train is running.

  As for the *size*, it would be advisable to include different kinds of objects having different dimensions (e.g., from small rocks to trucks) to properly study the performances of the model.

  Then, regarding *positioning*, we identified some "absolute" and some "relative" positions. By absolute positions we mean: i) obstacles on straights; ii) before, in the middle of, and after curves; iii) before, in the middle of, and after intersection points (e.g., level crossings); iv) before, in the middle of, and after infrastructures (e.g., tunnels, stations). On the other hand, by considering Fig. 4.3 as a visual support, with relative positions we mean objects: i) on the tracks occupying both the rails (Fig. 4.3(a)); ii) on the tracks occupying one rail only (Fig. 4.3(b)); iii) in the middle of the track without occupying any rail (Fig. 4.3(c)), i.e., the object is so small that it does not occupy rails but it could collide with the train's bogie/frame; iv) near to the tracks (Fig. 4.3(d)), i.e., the object does not physically obstruct the tracks but it could collide with the train's shape; v) objects pending from overhead supports (e.g., catenaries) which could collide with the train (Fig. 4.3(e)).

  It would be advisable that the dataset includes as many combinations as possible by taking into account the aforementioned "dimensions" (i.e., distance, size, absolute position, and relative position) to better emulate different real scenarios. Then, basing on these "dimensions", it would also be possible to carry out more structured model evaluations by analysing its performance in relation to, for example, "small objects; in the middle of the track without occupying any rail; on a straight; at a distance between 100 and 200 meters". Therefore, it would be also possible to realise omnicomprehesive matrices (similar to confusion matrices) to define which kind of risks can be effectively detected by the model and which not.

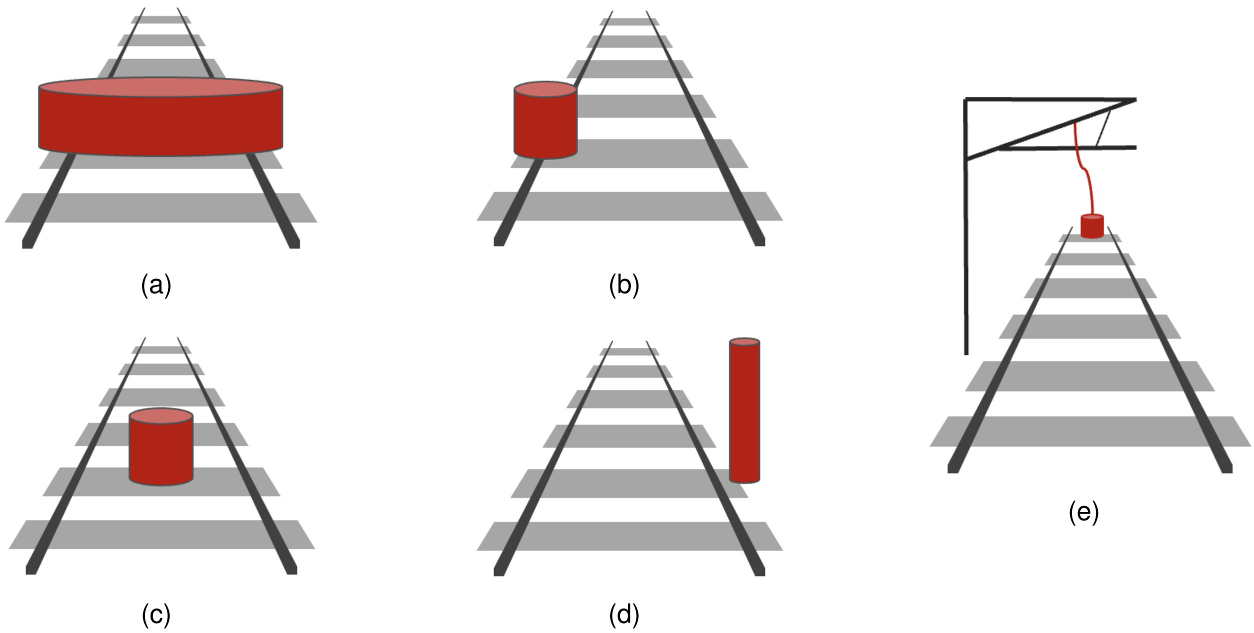


Fig. 4.3. Obstacles relative positions.

As indicated above, it would be advisable to train multiple instances of the same architecture (or different architectures) on different datasets to deal with the various possible conditions that characterise an open environment. Theoretically, it would be possible to handle different conditions with a singe model, however, there are two main aspects to consider:

- By splitting the problem into sub-problems, we are practically splitting the set of features that a specific model should learn facilitating its operations.
- By training different instances of the same architecture (or using different architectures) on different datasets, we are practically designing different systems. In our view, the data used to feed an AI architecture are an integral part of the model itself. Therefore, the same architecture trained on two different sets of data produces two different systems as the parameters of the model change with data. This will lead to a sort of systems redundancy and mitigate the single point-of-failure problem by (possibly) differentiating failure modes.

4.6. Artificial Intelligence and Machine Learning Models

In Section 4.4.1, we have proposed a framework (see Fig. 4.2) encompassing five modules each of which focus on a different sub-task as to efficiently address obstacle detection. Excluding the "Merge/Check Detections" module, which will trivially implement the logic to merge the two kinds (object and anomaly) of detection, for each of the other modules it is reasonable to adopt DL approaches. The main reason relies in the fact that DL approaches (e.g., CNNs and Autoencoders) have shown great results in respect to traditional ML approaches or other AI techniques when it comes to analysing image/video data [27].

By leveraging findings and results from the literature review of AI in railways performed within RAILS's WP1 [13], the transferability analysis performed in the first step of RAILS's WP2 [6], and the systematic review tailored on the topic of obstacle detection performed within the SMART project [8], we propose here a brief discussion oriented at identifying which, in our view, would be the "best practices" and most suitable AI approaches to implement the modules depicted in Fig. 4.2:

- **Rails Detection.** As from reference [8], various studies have been proposed in the literature to cope with the rail track detection task. They include approaches based on traditional Computer Vision (CV) techniques or DL. However, CV approaches (including geometry-based, Hough transform, etc) suffer the variation of light and weather conditions, complex backgrounds, and vibrations (e.g., if we consider a camera mounted in front of the train). Therefore, it would be advisable to adopt DL approaches. Among others, references [28] and [29] present two semantic segmentation solutions to detect tracks by leveraging, in our understanding, RGB images collected with a camera mounted in front of the train. Worth highlighting, from the KPI3 perspective, the network proposed in [28] runs at 20 frames per second (FPS) on a NVIDIA GTX1080, while the model proposed in [29] reaches 38 FPS (on a Titax X GPU). Lastly, for the sake of knowledge, reference [30] addressed the same task by leveraging data coming from thermal cameras. Therefore, given the promising results achieved by these studies, it would be advisable to exploit them or, at least, similar segmentation approaches based on, for example, autoencoders.
- **Object Detection.** The object detection task, from a general perspective, is one of the most investigated within many different sectors for disparate purposes (e.g., maintenance, obstacles detection) [6, 31], and railways is not an exception [8, 32, 33]. Therefore, it would be extremely time-consuming to develop a DL approach ex-novo when it would be possible to exploit already existing approaches (also from the rail sector [8]) or state-of-the-art architectures. What can be advisable, from the perspective of KPI3, is to adopt one-stage detectors as they are capable of processing images within a few milliseconds (e.g., YOLOv5[6]). However, they may lack in accuracy when compared to two-stage detectors such as R-CNN architectures (see [34] for example). Therefore, optimisations and a trade-off analysis should be carried out when practically addressing this task.
- **Anomaly Detection.** As far as we know, the detection of unknown objects is typically performed by means of other sensors rather than cameras (e.g., ALSTOM's obstacle detection system based on radar[7]). Then, AI and cameras are used to identify the type of the object. In our view, this could be one of the causes that limited work has been done towards image-based anomaly detection.

  Worth highlighting, the idea of detecting objects in a non-supervised manner is clearly not new. It has been already introduced within the GoSAFE RAIL project [7], where a subtraction method was presented to detect obstacles, and other studies such as [35]

[6]https://github.com/ultralytics/yolov5

[7]https://www.alstom.com/press-releases-news/2022/4/alstom-takes-another-step-towards-autonomous-train-operation-netherlands

(in railways) and [36, 37] (in automotive) where stereo images were leveraged, also in combination with DL, to detect obstacles.

However, despite image/video-based anomaly detection seems not to have been fully addressed yet in railways as in other sectors, we believe that the overall obstacle detection system presented in Fig. 4.2 could benefit from such a module both in terms of coverage (KPI2) and robustness. To that aim, it would be possible to start from approaches combining deep learning (especially autoencoders) and stereo vision, as mentioned above, or, from other studies focusing on detecting obstacles in an unsupervised manner leveraging a single camera [38, 39]. In the latter case, the high-level idea would be to train the network (e.g., the autoencoder) with images that do not contain obstacles and then, in case an image with an anomaly is presented in input, recognise and locate the anomaly. Another possibility would be to proceed by subtraction [40] by adopting a DL model to reconstruct the image with the anomaly and then compute the difference between the reconstructed image and the input image to obtain the anomaly map (see Fig. 4.4).

- **Distance Estimation.** This specific task has been recently analysed within the SMART project (DisNet [20, 41]). However, despite being innovative, it seems to be class-dependent, i.e., to estimate the distance it takes into account the dimensions of a specific object in different circumstances. This means that it would be possible to estimate the distance of objects in output to the object detection module. Differently, for what concerns anomalies, this approach could be considered as a suitable starting point, but adaptions seem to be required; otherwise, it would be possible to also leverage other solutions (e.g., [42]).

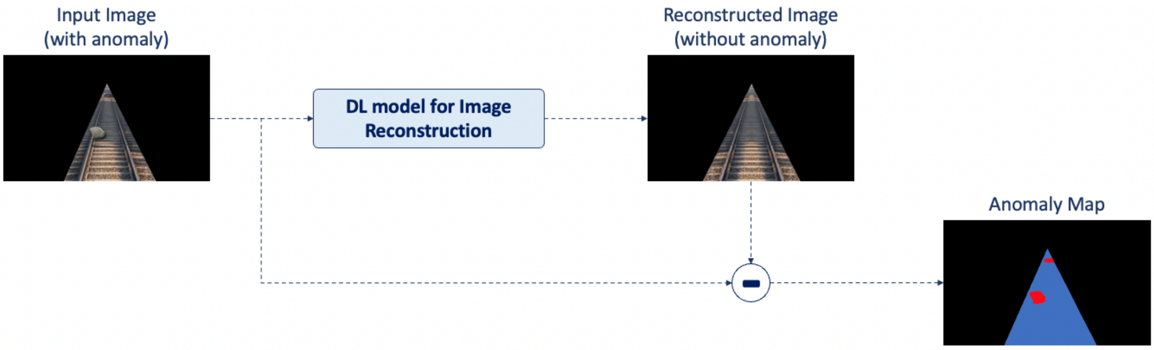


Fig. 4.4. High-Level Scheme for Anomaly Detection (excerpted from [40]).

Table 4.3 summarises the above and highlights the most promising classes of approaches that, in our view, should be taken into account to address the obstacle detection task. Column "contributions" reports a qualitative score (i.e., limited, medium, or high) indicating the coverage that, in our opinion and given the above analysis, the related topic has obtained within the literature or other projects.

Table 4.3: Suitable AI approaches

Module	Contributions	Suitable Approaches
Rails Detection	Medium	Semating Segmentation (e.g., based on Autoencoders).
Object Detection	High	Object Detectors (e.g., YOLO or R-CNN).
Anomaly Detection	Limited	Image Reconstruction (leveraging single cameras) or approaches computing disparity between stereo images (e.g., based on Autoencoders or CNN).
Distance Estimation	Limited	DNN, but also non-AI based approaches leveraging stereo images.

4.7. Expected Results and Possible Criticalities

In our view, the framework we propose (Fig. 4.2) could improve the robustness of camera-based obstacle detection systems. In real scenarios, such a system would most likely act as a sub-component of a larger system taking into account data from different sensors (e.g., radar, LiDAR, etc. [11]). However, it is clear that it would be better for each individual sub-system to be as robust as possible, especially when it comes to safety-critical applications. We expect that the AD module supports and extend the functioning of the OD module. In particular, the AD module would: i) confirm the detection of the OD module; ii) act as a fall-back system in case the OD module is faulty; and iii) detect anomalies that the OD module is not capable to detect. At the same time, also the OD module would support the AD one by confirming its detection, but this would be possible only for objects known a-priori. The main disadvantage is that this reasoning would only hold in the event that the two systems were completely trustworthy. However, by introducing a new system in the traditional path for obstacle detection (RD $\rightarrow$ OD $\rightarrow$ DE [8]), it is reasonable to think that the coverage (KPI2) and the robustness of the whole system would be increased. This is because we are introducing here a modular redundancy which encompasses two models implemented through different architectures trained in different modalities (supervised/unsupervised); hence, we would most likely have different failure modes. Unfortunately, also in this reasoning there is a drawback: we have only two systems for which, in addition, it is extremely difficult to estimate their reliability and trustworthiness. A possible solution would be to introduce, in the future, a thermal camera in addition to the RGB one. Then, it would be possible to replicate at least the AD module on thermal data as to obtain a third "independent" system and build what is commonly known as Triple Modular Redundancy (TMR) [43]. The RD module would remain a single point of failure, however, also in this case it would be possible to leverage thermal cameras to build another system to detect rail tracks.

4.7.1. Reliability and Trustworthiness Issues

Modular redundancy is a possibility to increment robustness, however, reliability concerns cannot be overcome by simply introducing multiple AI systems without taking into account their trustworthiness. Here, the main criticality is that the current state of development of intelligent technologies do not allow us to establish a precise safety integrity level for AI systems. Worth highlighting, looking to other sectors, practical guidelines and roadmaps are being developed for the safe adoption of ML (and AI in general); see, for example, the roadmap [44] and the latest report on the usable guidelines for Level 1 AI/ML (i.e., assistance to human) [45] from the European Union Aviation Safety Agency (EASA). Similarly, the European Commission recently published the "Trustworthy Autonomous Vehicles" [46] with the aim of analysing ALTAI requirements [47], and related criteria, for the assessment of the trustworthiness of autonomous vehicles with a particular focus, in our understating, on road transport vehicles. These works, together with other analyses specifically discussing requirements and safety level identification of AI systems in railways (e.g., those carried our it [48] for obstacle detection in autonomous train driving), represent valuable starting point for the definition of roadmaps and trustworthy specifications in railways.

However, nowadays, the **certification** of AI systems still remain the main open issue in the context of autonomous driving (or safety-critical applications in general) given, among others, the bespoken black-boxness and instability of many AI approaches [1, 13]. Explainable AI (XAI) solutions are expected to generate interpretable AI models and comprehensibly explain their functioning to human operators, however, they may also expose AI models to security threats (e.g., adversarial attacks [49]). Therefore, if on the one hand XAI approaches cold help in certifying AI systems, on the other hand, they must be reliable and certified in turn; worth mentioning, the eXplainable AI Working Group of the IEEE Computational Intelligence Society Standards Committee (CIS/SC) is currently working on the definition of standards for XAI [50].

4.7.2. Implementability Concerns

Besides these trustworthiness and reliability concerns, which would be clearly take into account in the following phases of the project, the purpose of our case study and related proof-of-concept would be to investigate the potentials of the proposed framework as to delineate guidelines and roadmaps for its implementation for generic obstacle detection tasks (e.g., intrusion detection in specific areas as well as on the tracks). In this context, we have identified some preliminary implementability criticalities that could affect the effectiveness of the approach.

First, the usage of camera sensors only may introduce some restrictions in terms of detectable obstacles and detection distance. Second, we have discussed about distance estimation and the approach proposed by SMART [41] to estimate the distance of known objects. However, concerns arises when it would come to estimate the distance of unknown objects (i.e., anomalies detected by the AD module) given the extremely limited work we have found in the literature in this direction. At the same time (third), it would also of extreme importance to estimate the dimensions of the anomalies. Detecting an object means also recognising its class, that will intrinsically bring with it information about the dimensions (e.g., the size of different cars would fall more or less in the same range); differently, when it comes to anomalies, we should introduce additional mechanisms to estimate the dimensions of the obstacle to effectively understand if it is a threat or not (e.g., a single leaf may not be a threat, but a leaf bench could reduce the friction between the train's wheels and the rails). Fourth, switch points may be critical as an object may occupy the track that will not be travelled by the train, hence, once an object has been recognised after a switch point, we should also understand if it represents a real threat for the train. Lastly, we also have some concerns related to both the approaches to be used for AD and hardware selection. Particularly, the selection of a suitable hardware is quite challenging; as to achieve real-time performances (KPI3), we should both optimise DL architectures and select the most suitable hardware taking into account both costs and performances.

All the concerns expressed above are summarised through the following questions:

- Can a camera-based obstacle detector effectively detect any kind of obstacles, or will there be obstacles (e.g., small objects) that would not be properly detected?

- How will the OD and AD modules perform with distant objects?
- How can we estimate the distance of an obstacle if its class in unknown?
- In case of AD, will it be possible to estimate the dimensions of the detected anomaly (i.e., of the obstacle)?
- In case of obstacles after a switch point, how can we recognise that the obstacle is lying on the train's path?
- Traditional background subtraction algorithms may be characterised by many true positives, i.e., they are prone to detect obstacles or "differences" even when they are not relevant. Can an approach based on DL mitigate this issue?
- In relation to the approaches used, what are the necessary characteristics that the hardware should have in order to achieve real-time performance?

In the following of the RAILS project, we will try to practically address some of these questions, with a specific focus on the AD module given that, as discussed in this chapter, limited work has been done so far, especially in the context of autonomous train driving.

5. Cooperative Driving for Virtual Coupling of Autonomous Trains

5.1. Introduction

The ever increasing demand in passenger and freight transportation is leading to the saturation of railway networks capacity, particularly on highly frequented lines. As a consequence, a lack of flexibility within the railway operations is emerging, with delays and overcrowding for passengers trains, or inefficiencies of transportation capacities for freight trains. An expansion of railway lines is not always possible due to the lack of space, high costs, and long times related to the building of additional infrastructure.

To overcome these limitations, the Shift2Rail MOVINGRAIL [51] and X2RAIL3 [52] programs proposed a novel paradigm in train control systems based on the concept of Virtual Coupling (VC). Its primary objectives are improving infrastructure utilisation and increasing capacity of existing railway lines. The idea, strictly related to the platooning concept in the automotive field, is to virtually couple two or more trains via Train-to-Train (T2T) communication, so that they can travel in formation with the same velocity while maintaining a desired inter-train safety distance among them [53]. Namely, on the basis of the shared information, each train, equipped with an on-board control system, should be able to automatically adapt its motion guaranteeing the tracking of a desired reference profile and maintaining a safe distance with respect to the preceding train [54]. This proof of concept investigates how and if the automotive platooning concept can be transferred to the railway domain. In particular, the aim is to explore the feasibility of VC in railways through AI techniques, in order to derive a qualitative and technology roadmap for the future VC deployment.

5.2. Background and Description

The most common national and international railway systems currently applied and in development are fixed blocks (ETCS Level 1 and 2 and most national signalling systems) and moving blocks (requiring ETCS Level 3). As stated in [52], both fixed and moving block approaches involve limitations to the potential line capacity, due to the absolute braking distance supervision: each train takes into consideration only its own braking characteristics to determine its permitted speed. As a consequence, the following trains are kept at a distance that is unnecessarily high, compared to the actual safety possibilities.

VC goes beyond the concept of moving block railway operations introduced by ETCS Level 3, allowing trains to be separated by a relative braking distance rather than by an absolute braking distance. The concept of Virtually Coupled Train Set (VCTS) implies a paradigm change called “braking the walls”: from absolute to relative braking distance (see Fig. 5.1).

It relies on a mutual exchange of relevant information between two or more following trains via T2T communication, in order to allow them to run at closer distance than the absolute braking distance of the rear consist. Trains running as virtually coupled should be capable to communicate to each other the set of data regarding their specific dynamic behaviour. In particular, the Master train shares information with the Radio Block Center (RBC) and receives a Movement Authority for the entire convoy of virtually coupled trains regarding the train separation with surrounding trains/convoys and the safe running on (interlocked)

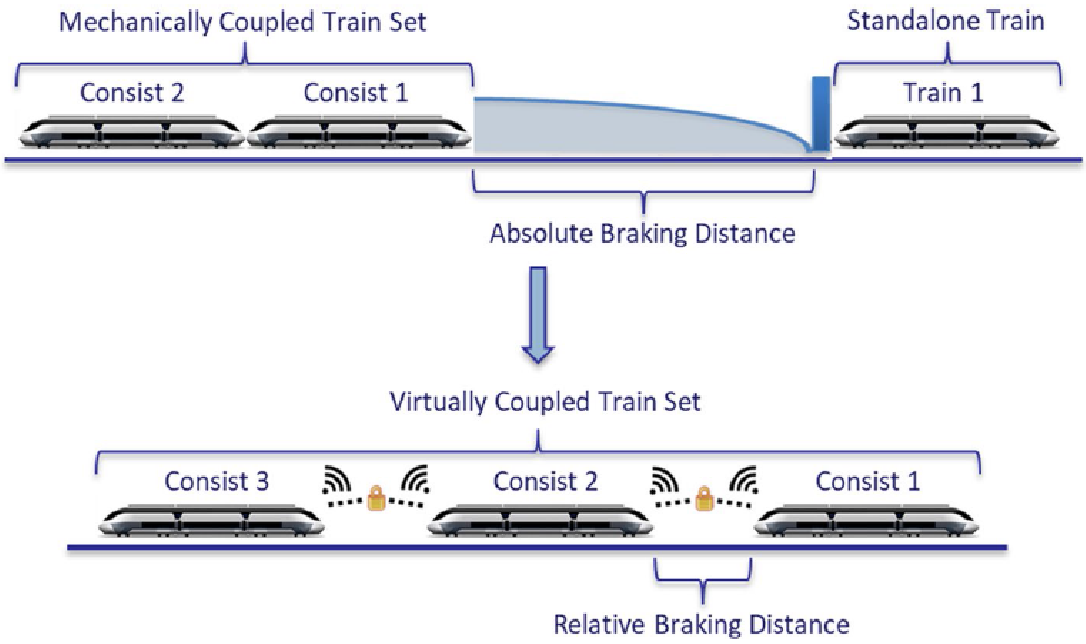


Fig. 5.1. “Braking the walls” new paradigm: from absolute to relative braking distance [55]

routes. The other trains in the convoy share inner-convoy information with the Master train and their adjacent trains, but they are not in contact with the RBC (or at least not in an active way). Thus, the RBC is used for the trackside train control of the entire convoy via the Master train only. The other trains are “slaves” to the Master train. On the basis of the shared information, the VCTS should be able to compute cooperative braking curves, which integrate the parameters related to the braking characteristics and status of the consists, as well as other parameters as communication network delay, exogenous factors, and so on. As a result, consists inside the platoon could be allowed to move as a convoy with a safe distance much lower than the braking distance needed for a full stop [56].

Given the relative short distances in between trains, VC also implies trains to be automatically driven by an automatic train operation to substantially reduce sight and reaction times of human drivers which would be unsafely long for this kind of operations.

The VC paradigm could bring a broader set of advantages over the traditional way to operate a railway network, such as, increase line capacity by reducing headways, increase operational flexibility, reduce costs, increase competitiveness making more efficient goods and passengers transportation with respect to road transportation [52].

However, the deployment of VCTS, despite its supposed benefits, requires a clearer understanding of its implications in term of feasibility, safety and actual capacity benefits. Research carried out so far in railway VC provides preliminary results and points out open challenges and critical issues. [57] introduced preliminary operational concepts for VC by defining an extended blocking-time model for comparing capacity occupation of VC with ETCS L3 moving-block operation. In the further work [58], a train-following model to describe train operations under VC and assess capacity performances under different operational settings has been developed. Capacity is measured in terms of space separation and time headway between consecutive trains. Simulation experiments show that VC significantly reduces train separation and time headways for all considered scenarios, compared to ETCS Level 2 and Level 3. This represents a promising result since VC could actually provide significant capacity improvements with respect to current railway systems.

The implications of VC in ERTMS/ETCS operational scenarios have also been investigated. [56] introduced the main advantages, current obstacles and future developments for the effective implementation of Virtual Coupling withing the ERTMS standard specification. In [53], a proof of concept has been proposed by introducing a specific operating mode within the ERTMS/ETCS standard, and a coupling control algorithm accounting for time-varying delays affecting the communication links has been addressed.

The identification of technical performance requirements for VC communication (direct T2T over short and long distances, low latency, etc.) has been addressed in MOVINGRAIL Deliverable D3.3 [59]. Specifically, it discloses that the communications architecture for VC should be based around 5G principles with a cellular network connection for long distance communication and a Peer to Peer direct link similar to IEEE802.11 (Wi-Fi), but fully integrated into 5G, for short range communication.

Available studies have only partially touched upon the effects of risk factors on safety and capacity. [60] proposed a SWOT analysis for different railway market segments to assess feasibility of VC and investigate the applicability of such a concept, pointing out the safety, operational, and technological challenges that need to be carefully addressed. [56] and [53], too, highlighted the need of investigating the safety-related impact of VC against hazardous scenarios and critical failures. In this direction, [61] introduced the concept of dynamic safety margin for VCTS to dynamically adjust train separations so to always keep required safety distances when hazardous operational events occur. The aim is to take into account relevant risk factors in real-life operations like: T2T communication delays, extended driving reaction times, train positioning errors and emergency braking applications. Indeed, when considering those factors, the train separation under VC needs to be increased by additional safety margins to remove any safety risks arising from their individual or combined presence.

Although the VC concept has been widely explored from conceptual point of view or via simulations, there are still few works so far focused on the development of suitable control systems to be equipped on board. About this gap, for instance, interesting results are presented in [58], where authors captures multiple states and their VC train operation transitions, while safety issue are analysed in [62], providing also several monitoring theorems. Instead, based on the developments in platooning of autonomous vehicles, different control strategies have been proposed in the technical literature for realising VC. Specifically, an optimal Model Predictive Control approach has been leveraged in [63] to guarantee a safe distance between two consecutive trains under state/input constraints, or in [64] for metro lines. The same strategy, developed in a centralised fashion, has been applied in [65] to consider the effects of the gradient on the braking distance. Instead, a Distributed Model Predictive Control (DMPC) is suggested in [66] to consider the coupled constraint of safety braking distance and the individual constraints of speed limit variations and restricted traction/braking performance.

It is worth noting that, besides the model-based control techniques mentioned above, model-free and Deep Reinforcement Learning (DRL) controllers are spreading in the ITS field. First attempts in the cooperative driving of autonomous connected vehicles can be found in [67, 68], where the platooning problem is solved via a Deep Deterministic Policy Gradient (DDPG) and a Hybrid DRL approach, respectively. With respect to railway systems, and more specifically to train convoys, DRL strategies are explored, e.g., for improving the train timetable rescheduling/routing in [69] or for controlling a single train in energy-efficient way in [70]. However, to the best of our knowledge, the possibility of applying the DRL concept for the VCTS paradigm is not investigated yet.

5.2.1. VCTS Functional Layer Structure

The VCTS system should be able to provide various functionalities which can be grouped into different classes. In the concept proposed in [71], these classes are organised in a vertical layer structure, and are characterised by distinct levels of abstraction: from a macroscopic view of the whole railway network to the microscopic movements of single convoys. Four functional layers and their interfaces are defined, interacting with each other or external actor (see Fig. 5.2):

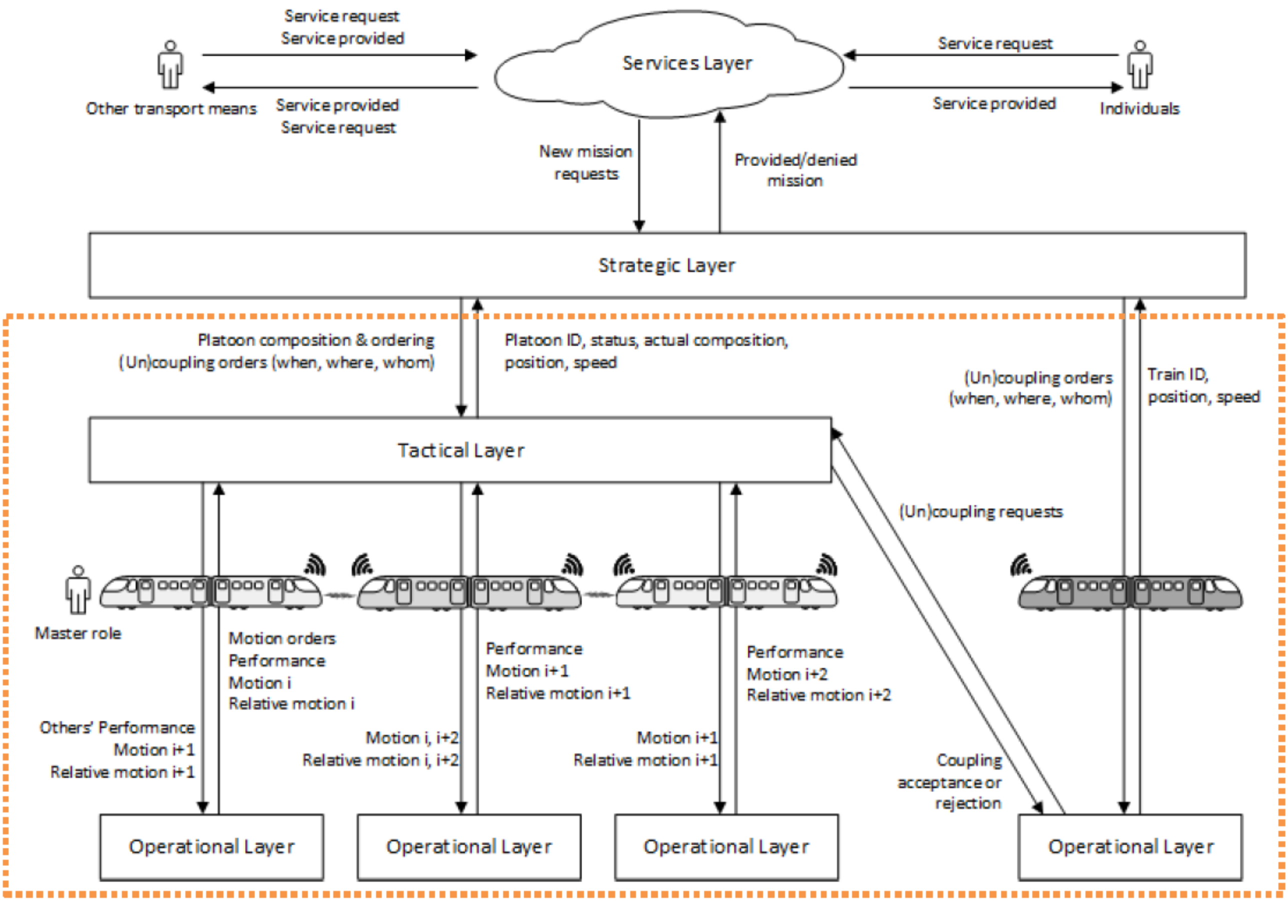


Fig. 5.2. VCTS functional layers [52]

- The service layer is integrated in the Mobility-as-a-Service (MaaS) platform and asks for new and ongoing missions.
- The strategic layer, following the request of the service layer, defines the platoons, their composition and ordering, based on compatibility, destinations and schedules. It also defines when and where the trains are joining and leaving the platoon.
- The tactical layer coordinates the actual platoon movements and manoeuvres from the instance a joining request is received until the platoon is dissolved. It manages unexpected events and sudden degraded modes of any of the platoon units.

- The operational layer, implemented on each train, manage the local control of each consist and ensure the safe execution of the commands from the tactical layer. The safety-critical functions of VCTS on the train level are allocated on this layer.

In the following, we will refer to railway VC from a tactical layer perspective. The VC control strategy, indeed, is embedded in the tactical layer, which is in charge of managing the VCTS by defining the speed and acceleration targets, as well as the headway between trains. As a consequence, line capacity strongly depends on the effectiveness of the tactical layer functionalities.

5.3. Objectives and Research Questions

The main objectives of the VC paradigm are improving infrastructure utilisation, increasing the capacity of the existing railway lines (reduction of trip times, headways, etc.) with respect to current systems, as well as increasing flexibility allowing platooning among trains of different types. The tactical layer is responsible of the required distance between trains that guarantees safe braking to standstill. In an ideal world without any delays, uncertainties and inaccuracies, VC objectives would be optimised, as the relative braking distance would be zero. Indeed, in this case, trains could travel at the same speed, and thus follow the same braking curve never changing the distance between them [72]. However, in reality, railway environments are characterised by multiple factors and uncertainties that cause deviation from this ideal behaviour, requiring additional safety margins and thus affecting the efficiency of VC. These can be:

- reaction delay, that is, the elapsed time between brake application of two trains, regardless the reason for the delay;
- latencies in T2T communication;
- heterogeneous trains with different operational performances (e.g., different braking capabilities, different speed categories);
- trains with variable mass (e.g., freight trains);
- track conditions (e.g., adhesion factors, gradients);
- exogenous factors (e.g., weather conditions);
- uncertainties in train location information.

When these effects are combined in real-life scenarios, they can significantly affect the performances of the tactical layer.

In view of this, some research questions arise:

- **RQ1** Can we transfer to railway VC some of the methods already exploited for vehicle platooning?
- **RQ2** Can AI approaches be leveraged to ensure the effectiveness of the tactical layer functionalities both in nominal and uncertain environments?
- **RQ3** Can we address possible answers to RQ1 and RQ2 through a proof of concept in order to inspire future developments and a technology roadmap?

To give an answer to the above questions, and thus evaluate the effectiveness of the VCTS paradigm, some evaluation criteria should be considered. In particular, some risk assessment indexes could be transferred from vehicle platooning, as, for instance, time gap and time to collision [73], which are two typical rear-end collision indexes. Furthermore, other performance indicators already exploited in automotive could be transferred to the railway domain, as energy consumption [74] or tracking error [75]. Indeed, as already pointed out in the previous section, some of the first studies on VCTS (see for instance [58]) actually exploits some KPIs which are well established in automotive. In view of this, in Table 5.1 we propose some KPIs derived from the automotive which could be well adapted and transferred to the railway sector, and thus can be used to evaluate the performances of railway VC.

Table 5.1: KPIs to evaluate the effectiveness of VCTS paradigm

KPI	Description
KPI1	**Time gap**, i.e., the time interval between two consecutive consists
KPI2	**Time to collision**, which is the remaining time before a rear-end accident, assuming unchanged speeds of both consists
KPI3	**Trip time**, that is, the time needed to accomplish a given mission
KPI4	**Tracking error**, i.e., the error between the desired reference behaviour and the actual one
KPI5	**Energy consumption**, used to evaluate the energy efficiency

5.4. Methodology and Tools

In the automotive field, most of the current autonomous driving decision making systems are focusing on model-based approaches, which requires a prior knowledge of the environment and vehicle characteristics to manually design a driving policy. Thus, traditional controllers make use of an a priori model composed of fixed parameters. However, in complex environments, they cannot foresee every possible situation that the system has to cope with.

Recent advances in Machine Learning (ML) enables the possibility for learning based approaches for autonomous driving decision making. Learning controllers make use of training information to learn their models over time. With every gathered batch of training data, the approximation of the true system model becomes more accurate, thus enabling model flexibility, consistent uncertainty estimates and anticipation of repeatable effects and disturbances that cannot be modelled before deployment [76]. Among ML methods, DRL is a goal-oriented learning tool wherein the agent or decision maker learns a policy to optimise a long-term reward by interacting with the environment. At each step, an RL agent gets evaluative feedback about the performance of its action, allowing it to improve the performance of subsequent actions.

As already highlighted in Deliverable D2.1 of RAILS WP2 [6], few contributions regarding AI approaches for vehicle platooning have been proposed so far. However, they show promising results when RL methods are exploited. [77] proposed a RL optimal controller for the vehicle platooning problem based on a DDPG algorithm. The results have been compared with a conventional Model Predictive Control (MPC) strategy. Simulations confirm that the RL controller outperform the MPC in terms of computational time and control effort specially in more realistic and complex scenarios, while maintaining similar root mean square error in the inter-vehicle distances. [5] compared DRL based on DDPG and traditional MPC for Adaptive Cruise Control (ACC) design in car-following scenarios. Simulations show that, when there are no modelling errors and the testing inputs are within the training data range, the DRL solution is equivalent to MPC with a sufficiently long prediction horizon. The DRL control performance degrades when the testing inputs are outside the training data range; however, when there are modelling errors due to control delay, disturbances, or uncertainties, the DRL-trained policy performs better when the modelling errors are large while having similar performances as MPC when the modelling errors are small.

From the results in automotive, it emerges that RL methods based on DDPG can outperform, especially in complex and uncertain scenarios, conventional model-based approaches. Since the railway environment, too, includes so many factors and uncertainties which could affect the performances of railway VC (as mentioned in the previous section), it seems that DDPG-based RL approaches could be preferred to conventional model-based control algorithms for the deployment of the tactical layer functionalities, hence addressing **RQ1**. Namely, they could be capable to adapt the optimal solution in real-time considering all current uncertainties and contingencies without any prior knowledge of the railway environment (see **RQ2**).

In view of these considerations, we propose a proof of concept to explore the technical feasibility of railway VC through RL methods based on DDPG. To the best of our knowledge, no AI solutions have been addressed so far to tackle VC control strategies in railways. Hence, a possible technology roadmap for VCTS feasibility could be defined, therefore addressing **RQ3**.

5.5. Artificial Intelligence and Machine Learning Models

On the basis of the considerations made above, we aim at investigating how the tactical layer functionalities could be deployed through an RL control algorithm based on DDPG. It is a model-free actor-critic method able to learn a competitive policy in a continuous action space using states in the designed observation space [78]. In other words, DDPG allows to train an agent interacting with an unknown environment (without any prior knowledge of the latter) via a *learn-by-doing* process: the RL agent learns, through trial and error, the best way to accomplish a task, so that the trained agent becomes able to adapt and react to the surrounding environment.

The actor-critic exploits two different RL methods: the value-based RL approach, based on Deep Q-learning Network (DQN), which approximates the nonlinear observation state-action value and allows the evaluation of the quality of the chosen action, and a policy method based on Deep Policy Gradient (DPG), which bypasses the evaluation of the quality value via a direct estimation of the relation between the observation state and the action.

Fig. 5.3 shows a general architecture of a DDPG-based control algorithm, considering four different Deep Neural Networks (DNN), namely: actor, critic, target actor and target critic. The actor (DPG-based) aims to compute the control inputs via the estimation of the competitive policy, while the critic (DQN-based) *"criticises"* the action chosen by the actor. The target networks are exploited to stabilise the training process and present the same structure of the corresponding main networks. Indeed, in order for a RL agent to learn a given task, a training process is necessary. This is based on a reward function, which gives an evaluation of the quality of the chosen action. In particular, the policy inherited in the agent itself is updated maximising a cumulative reward obtained by interacting with the environment. Hence, the reward is the feedback that carries the information of the adequacy of the performance based on the agent's policy.

In view of this, to design a VCTS control algorithm based on DDPG, the following features should be properly defined: the action space, the observation space, and the reward function (see Fig. 5.4 for a conceptual view of a DDPG-based VCTS). Specifically, the continuous action space represents the control input, e.g., the acceleration/deceleration of the consists at each time step. The observation space is composed of all the observable states which allow a perception of the scenario. In designing the observation space, the following information should be considered: the desired reference coming from the RBC through T2I communication, train positions and velocities coming from T2T communication (according to the communication topology), some train characteristics, such as train length, the energy consumption, and so on.

An accurate shaping of the reward function is necessary to take into account the main objectives of the VCTS paradigm. In particular, in order to enhance safety and improve railway capacity and performances at the same time, the reward function could take into account some of the risk assessment KPIs defined in Table 5.1, such as time gap or time to collision, as well as other objectives, such as energy consumption, or even comfort (for an RL-based ACC controller, a suitable reward function has been proposed in [79]). However, the convergence of the DDPG algorithm to the optimal solution is strictly related to the accuracy of the reward function: the more the objectives considered in the reward function, the more sub-optimal the solutions that could be achieved.

During the learning process, the RL agent learns to react to the unknown surrounding environment, which embeds all the unpredictable and uncertain factors characterising railway dynamic and complex scenarios, such as track conditions, exogenous factors, and so on. Note that the consists themselves and their operational performances are embedded in the environment. This means that the proposed approach could allow the coupling of trains with different operational capabilities, ensuring platooning among heterogeneous trains while absorbing uncertainties arising in real driving conditions. It is worth highlighting that, in order to ensure the effectiveness of VCTS, the technical performance requirements for T2T communications (direct T2T over short and long distances, low latency, etc.) should be satisfied. As emerged from [51], 5G technology could meet the T2T communication requirements for VCTS.

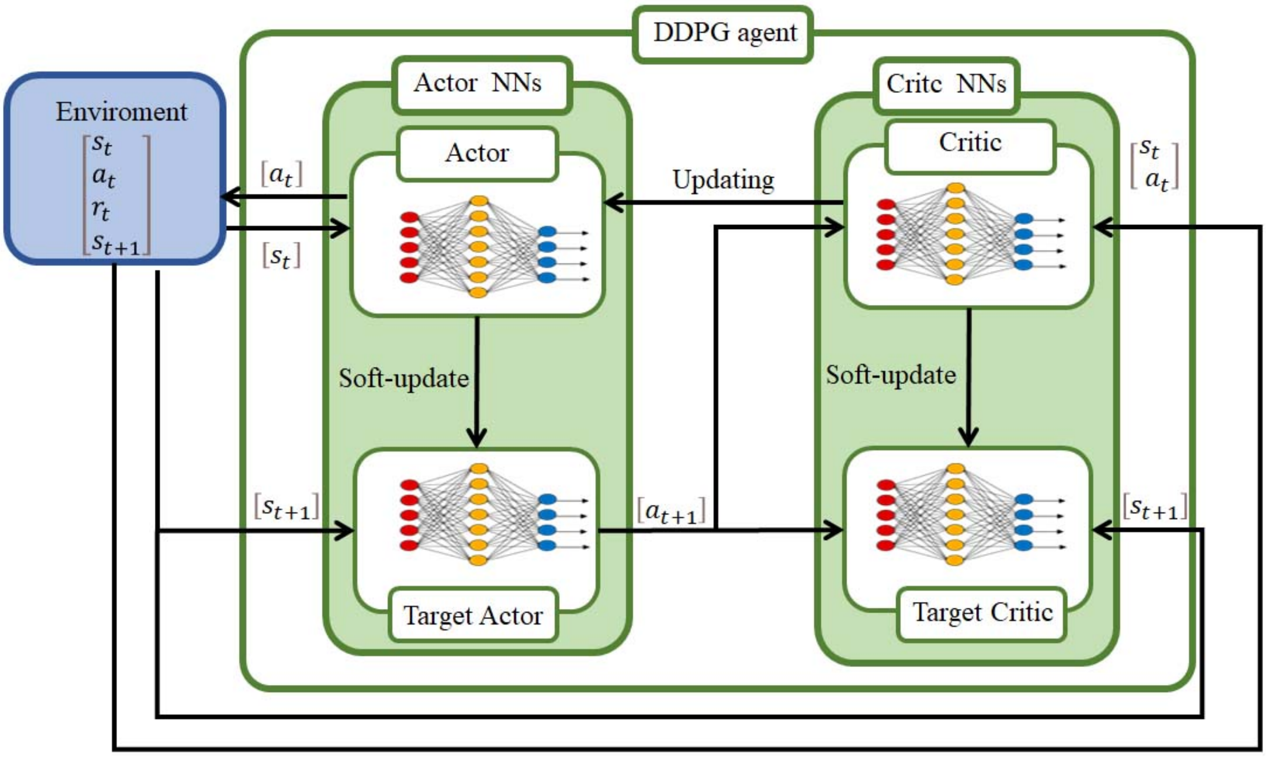


Fig. 5.3. General architecture of a DDPG-based algorithm (a_t = action space at time t, s_t = observation space at time t, r_t = reward function at time t. Note that both the action and observation spaces are updated at each time step)

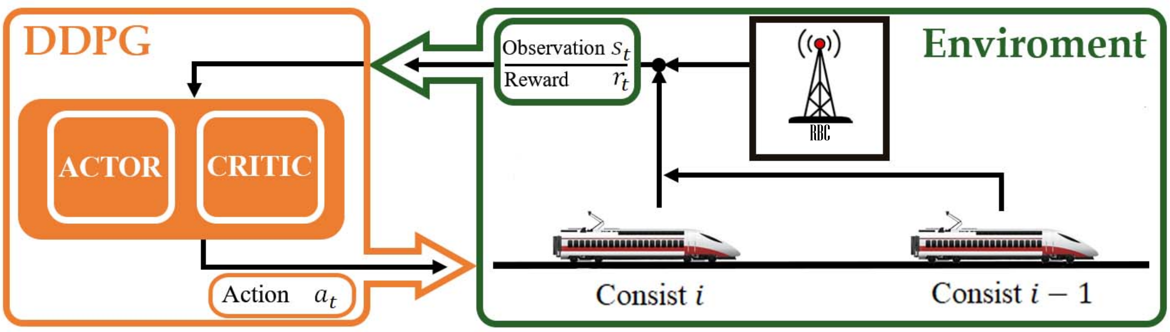


Fig. 5.4. Conceptual view of DDPG-based RL approach for VC in railways

5.6. Reference Datasets

Differently from other ML algorithms, as supervised and unsupervised learning, which exploit defined and fixed training datasets [80], the DRL methods work in high-dimensional, continuous action spaces, to deal with notably physical control tasks [78]. Such large action spaces are difficult to explore efficiently, and any datasets would be inadequate for the purpose. That's why, as also emerges from the automotive field, (see Deliverable D2.1 of RAILS WP2 [6]), simulators for virtual validation and training are required. There are several advantages for using simulators as a training tool for RL. The first is that one can afford many more samples, since simulations can be significantly faster and cheaper than the real experiments. The second is safety, which cannot be guaranteed in real scenarios for trial-and-error learning of RL, especially in worst case situations [81].

However, to the best of our knowledge, even though some simulators are currently available for railway virtual testing and RL training (see for instance SUMO [82] and Anylogic [83]), they do not allow the possibility of considering the VCTS paradigm, which also requires to take into account T2T communications. This represents a critical aspect for the potential exploitation of RL methods for the actual deployment of railway VC. In view of this, an ad-hoc simulator should be provided for the validation and training of the proposed approach. This simulator should take into account different operative scenarios, train characteristics, track conditions, and exogenous factors, to be exploited during the training phase. Note that there is still a gap to be filled between the accuracy of a simulated world and real-world scenarios [84], and this could affect the performance of learning systems.

5.7. Expected Results and Possible Criticalities

To assess the feasibility and to evaluate the performances of RL-based VCTS, some evaluation criteria should be considered. In particular, some of the KPIs proposed in Table 5.1 could be exploited to compare the performances of the proposed RL approach with those of a traditional model-based control strategy, as well as of current fixed and moving block railway systems, considering different operational scenarios, uncertain environments, heterogeneous train sets, exogenous factors, etc.

As it follows from the automotive field, it is expected that the performances of a DDPG-based RL algorithm could overcome those of a VC model-based approach, especially in complex and uncertain environments. Furthermore, the first evidences in railway VC already shared better performances of VCTS with respect to current fixed and moving block systems; thus, we look forward to confirm those results through the exploitation of AI techniques.

As already pointed out, possible criticalities for using DDPG-based RL methods for the deployment of VTCS paradigm could be represented by:

- the shaping of the reward function: to improve the performance of the proposed control strategy, a suitable reward function has to be designed. It could be achieved, for instance, after testing a variety of reward combinations (see [85]);
- the training process: a suitable railway simulator should be provided for the validation and the training phase, since the robustness/resilience of the proposed RL method to the unmodelled and uncertain factors affecting the railway environment can be improved via a correct and deep training process.

It is worth to highlight that the proposed methodology is very challenging, since the usage of AI models in safety-critical systems is still an open debate, and efforts are being made to bring AI and functional safety closer to each other [84].

This proof of concept is intended to give a first insight towards the definition of a qualitative and technology roadmap which could lead to the deployment of AI applications aiming at enhancing rail safety and automation.

In this direction, different alternatives to the proposed approach could be investigated. For instance, a Cooperative Adaptive Cruise Control (CACC) strategy based on a Multi-Agent RL (MARL) algorithm, which should be adapted to the VCTS paradigm, may be considered for future research. Concepts and examples on CACC in the automotive field can be found in [86–95] and reference therein.

6. Conclusions

In this deliverable, we have addressed the definition of two relevant case-studies in the context of AI for autonomous and cooperative driving in railways. The case-studies will be developed in the continuation of the project in order to get some useful insights to support the definition of future roadmaps in the use of AI approaches in related railway applications. At this stage, the case-studies have been specified mainly in terms of background and motivation, objectives and research questions, planned AI methodologies and tools, available datasets, AI/ML models to be used, expected results and possible criticalities. We expect that actual experimentation during the proof-of-concept stage will further highlight a set of opportunities and challenges to be addressed by researchers and engineers in order to leverage on the full potential of AI and ML.

Due to the project scope, a selection of relevant scenarios had to be made considering the planned efforts, limited time requirements, and results achievable; however, we highlighted a comprehensive set of research directions and perspectives that could be explored in addition to the two case-studies described in this deliverable.

Bibliography

[1] F. Flammini, L. De Donato, A. Fantechi, and V. Vittorini, "A vision of intelligent train control," *Preprint submitted to International Conference on Reliability, Safety, and Security of Railway Systems (RSSRail 2022), Lecture Notes in Computer Science, Springer*, 2022.

[2] C. Di Meo, M. Di Vaio, F. Flammini, R. Nardone, S. Santini, and V. Vittorini, "Ertms/etcs virtual coupling: Proof of concept and numerical analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2545–2556, 2020.

[3] P. Koopman and M. Wagner, "Toward a framework for highly automated vehicle safety validation," SAE Technical Paper, Tech. Rep., 2018.

[4] A. Moujahid, M. E. Tantaoui, M. D. Hina, A. Soukane, A. Ortalda, A. ElKhadimi, and A. Ramdane-Cherif, "Machine learning techniques in adas: a review," in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. IEEE, 2018, pp. 235–242.

[5] Y. Lin, J. McPhee, and N. L. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 221–231, 2020.

[6] RAILS, "Deliverable D2.1: WP2 Report on case studies and analysis of transferability from other sectors (safety and automation)," 2021. [Online]. Available: https://rails-project.eu/downloads/deliverables

[7] GoSAFE RAIL, "Deliverable D1.3: Report on monitoring system for obstruction detection," Tech. Rep., 2018.

[8] D. Ristić-Durrant, M. Franke, and K. Michels, "A review of vision-based on-board obstacle detection and distance estimation in railways," *Sensors*, vol. 21, no. 10, p. 3452, 2021.

[9] GoSAFE RAIL, "Global Safety Management Framework for RAIL Operations," 2016-2019, http://www.gosaferail.eu/.

[10] SMART, "Smart Automation of Rail Transport," 2016-2019, http://smart.masfak.ni.ac.rs.

[11] SMART2, "Advanced integrated obstacle and track intrusion detection system for smart automation of rail transport," 2019-2022, https://smart2rail-project.net.

[12] L. De Donato, F. Flammini, S. Marrone, C. Mazzariello, R. Nardone, C. Sansone, and V. Vittorini, "A survey on audio-video based defect detection through deep learning in railway maintenance," *IEEE Access*, pp. 1–1, 2022.

[13] RAILS, "Deliverable D1.2: Summary of existing relevant projects and state-of-the-art of AI applications in railways," 2021, DOI: 10.13140/RG.2.2.11353.03686. [Online]. Available: https://rails-project.eu/downloads/deliverables

[14] P. Hyde, C. Ulianov, J. Liu, M. Banic, M. Simonovic, and D. Ristic-Durrant, "Use cases for obstacle detection and track intrusion detection systems in the context of new generation of railway traffic management systems," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 236, no. 2, pp. 149–158, 2022.

[15] M. Martinez, C. Sitawarin, K. Finch, L. Meincke, A. Yablonski, and A. Kornhauser, “Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars,” *arXiv preprint arXiv:1712.01397*, 2017.

[16] B. Hurl, K. Czarnecki, and S. Waslander, “Precise synthetic image and LiDAR (presil) dataset for autonomous vehicle perception,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2522–2529.

[17] A. Karovalia. (2020) Building a self-driving vehicle in GTA-V using deep learning and convolutional neural network. Accessed: 24 May 2022. [Online]. Available: https://alzaibkarovalia.medium.com/building-a-self-driving-vehicle-in-gta-v-using-deep-learning-and-convolutional-neural-network-696b38b4c81e

[18] B. Guo, G. Geng, L. Zhu, H. Shi, and Z. Yu, “High-speed railway intruding object image generating with generative adversarial networks,” *Sensors*, vol. 19, no. 14, p. 3075, 2019.

[19] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, and C. Beleznai, “Railsem19: A dataset for semantic rail scene understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[20] D. Ristić-Durrant, M. A. Haseeb, M. Franke, M. Banić, M. Simonović, and D. Stamenković, “Artificial intelligence for obstacle detection in railways: Project smart and beyond,” in *Dependable Computing - EDCC 2020 Workshops*, S. Bernardi, V. Vittorini, F. Flammini, R. Nardone, S. Marrone, R. Adler, D. Schneider, P. Schleiß, N. Nostro, R. Løvenstein Olsen, A. Di Salle, and P. Masci, Eds. Cham: Springer International Publishing, 2020, pp. 44–55.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[23] M. J. Pappaterra, F. Flammini, V. Vittorini, and N. Bešinović, “A systematic review of artificial intelligence public datasets for railway applications,” *Infrastructures*, vol. 6, no. 10, p. 136, 2021.

[24] X. Li, K. Kou, and B. Zhao, “Weather gan: Multi-domain weather translation using generative adversarial networks,” 2021.

[25] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[26] C.-T. Lin, S.-W. Huang, Y.-Y. Wu, and S.-H. Lai, “Gan-based day-to-night image style transfer for nighttime vehicle detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 951–963, 2021.

[27] RAILS, “Deliverable D1.1: Definition of a reference taxonomy of AI in railways,” 2020, DOI: 10.13140/RG.2.2.24887.75681. [Online]. Available: https://rails-project.eu/downloads/deliverables

[28] Y. Wang, L. Wang, Y. H. Hu, and J. Qiu, “Railnet: a segmentation network for railroad detection,” *IEEE Access*, vol. 7, pp. 143 772–143 779, 2019.

[29] Z. Wang, X. Wu, G. Yu, and M. Li, "Efficient rail area detection using convolutional neural network," *IEEE Access*, vol. 6, pp. 77 656–77 664, 2018.

[30] M. Wedberg, "Detecting rails in images from a train-mounted thermal camera using a convolutional neural network," 2017.

[31] RAILS, "Deliverable 3.1 – WP3 Report on case studies and analysis of transferability from other sectors (predictive maintenance and defect detection)," 2021. [Online]. Available: https://rails-project.eu/downloads/deliverables

[32] L. De Donato, F. Flammini, S. Marrone, C. Mazzariello, R. Nardone, C. Sansone, and V. Vittorini, "A survey on audio-video based defect detection through deep learning in railway maintenance," *IEEE Access*, pp. 1–1, 2022.

[33] R. Tang, L. De Donato, N. Besinović, F. Flammini, R. M. Goverde, Z. Lin, R. Liu, T. Tang, V. Vittorini, and Z. Wang, "A literature review of artificial intelligence applications in railway systems," *Transportation Research Part C: Emerging Technologies*, vol. 140, p. 103679, 2022.

[34] D. He, Y. Qiu, J. Miao, Z. Zou, K. Li, C. Ren, and G. Shen, "Improved mask r-cnn for obstacle detection of rail transit," *Measurement*, vol. 190, p. 110728, 2022.

[35] H. Mukojima, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, M. Ukai, N. Nagamine, and R. Nakasone, "Moving camera background-subtraction for obstacle detection on railway tracks," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3967–3971.

[36] V. D. Nguyen, H. Van Nguyen, D. T. Tran, S. J. Lee, and J. W. Jeon, "Learning framework for robust obstacle detection, recognition, and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1633–1646, 2017.

[37] A. Dairi, F. Harrou, M. Senouci, and Y. Sun, "Unsupervised obstacle detection in driving environments using deep-learning-based stereovision," *Robotics and Autonomous Systems*, vol. 100, pp. 287–301, 2018.

[38] A. Boussik, W. Ben-Messaoud, S. Niar, and A. Taleb-Ahmed, "Railway obstacle detection using unsupervised learning: An exploratory study," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 660–667.

[39] T. Ohgushi, K. Horiguchi, and M. Yamanaka, "Road obstacle detection method based on an autoencoder with semantic segmentation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[40] L. Wang, D. Zhang, J. Guo, and Y. Han, "Image anomaly detection using normal data only by latent space resampling," *Applied Sciences*, vol. 10, no. 23, p. 8660, 2020.

[41] D. Ristić-Durrant, M. Franke, K. Michels, V. Nikolić, M. Banić, and M. Simonović, "Deep learning-based obstacle detection and distance estimation using object bounding box," *Facta Universitatis. Series: Automatic Control and Robotics*, vol. 20, no. 2, pp. 075–085, 2021.

[42] S. Emani, K. Soman, V. Sajith Variyar, and S. Adarsh, "Obstacle detection and distance estimation for autonomous electric vehicle using stereo vision and dnn," in *Soft Computing and Signal Processing*. Springer, 2019, pp. 639–648.

[43] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM journal of research and development*, vol. 6, no. 2, pp. 200–209, 1962.

[44] EASA, "Artificial Intelligence Roadmap - A human-centric approach to AI in aviation," 2020. [Online]. Available: https://www.easa.europa.eu/document-library/general-publications/easa-artificial-intelligence-roadmap-10
[45] ——, "EASA Concept Paper: First usable guidance for Level 1 machine learning applications," 2021. [Online]. Available: https://www.easa.europa.eu/newsroom-and-events/news/easa-releases-its-concept-paper-first-usable-guidance-level-1-machine-0
[46] European Commission and Joint Research Centre, *Trustworthy autonomous vehicles : assessment criteria for trustworthy AI in the autonomous driving domain*, 2021.
[47] AI HLEG, "The Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment, European Commission, Brussels," 2020.
[48] S. Rosić, D. Stamenković, M. Banić, M. Simonović, D. Ristić-Durrant, and C. Ulianov, "Analysis of the safety level of obstacle detection in autonomous railway vehicles," *Acta Polytechnica Hungarica*, vol. 19, no. 3, 2022.
[49] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.
[50] Standard for XAI – eXplainable Artificial Intelligence - for Achieving Clarity and Interoperability of AI Systems Design. [Online]. Available: https://development.standards.ieee.org/myproject-web/public/view.html#pardetail/8923
[51] MOving block and VIrtual coupling New Generations of RAIL signalling (MOVINGRAIL). [Online]. Available: https://projects.shift2rail.org/s2r_ip2_n.aspx?p=MOVINGRAIL
[52] Advanced Signalling, Automation and Communication System (IP2 and IP5) – Prototyping the future by means of capacity increase, autonomy and flexible communication (X2RAIL3). [Online]. Available: https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-3
[53] C. Di Meo, M. Di Vaio, F. Flammini, R. Nardone, S. Santini, and V. Vittorini, "Ertms/etcs virtual coupling: Proof of concept and numerical analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2545–2556, 2019.
[54] J. Park, B.-H. Lee, and Y. Eun, "Virtual coupling of railway vehicles: Gap reference for merge and separation, robust control, and position measurement," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
[55] M. Schenker, "S2r innovation days presentation x2r3-td2. 8 virtually coupled train sets," 2020.
[56] F. Flammini, S. Marrone, R. Nardone, A. Petrillo, S. Santini, and V. Vittorini, "Towards railway virtual coupling," in *2018 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*. IEEE, 2018, pp. 1–6.
[57] E. Quaglietta, "Analysis of platooning train operations under v2v communication-based signaling: Fundamental modelling and capacity impacts of virtual coupling," in *Proceedings of the 98th Transportation Research Board Annual Meeting*. Transportation Research Board (TRB), 2019.
[58] E. Quaglietta, M. Wang, and R. M. Goverde, "A multi-state train-following model for the

analysis of virtual coupling railway operations," *Journal of Rail Transport Planning & Management*, vol. 15, p. 100195, 2020.

[59] "MOVINGRAIL Deliverable D3.3 - Proposals for Virtual Coupling Communication Structures," 2020. [Online]. Available: https://projects.shift2rail.org/s2r_ip2_n.aspx?p=MOVINGRAIL

[60] J. Aoun, E. Quaglietta, and R. M. Goverde, "Investigating market potentials and operational scenarios of virtual coupling railway signaling," *Transportation Research Record*, vol. 2674, no. 8, pp. 799–812, 2020.

[61] E. Quaglietta, P. Spartalis, M. Wang, R. M. Goverde, and P. van Koningsbruggen, "Modelling and analysis of virtual coupling with dynamic safety margin considering risk factors in railway operations," *Journal of Rail Transport Planning & Management*, vol. 22, p. 100313, 2022.

[62] Y. Zhang and H. Wang, "Topological manifold-based monitoring method for train-centric virtual coupling control systems," *IET Intelligent Transport Systems*, vol. 14, no. 2, pp. 91–102, 2020.

[63] J. Felez, Y. Kim, and F. Borrelli, "A model predictive control approach for virtual coupling in railways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2728–2739, 2019.

[64] Z. Wu, C. Gao, and T. Tang, "A virtually coupled metro train platoon control approach based on model predictive control," *IEEE Access*, vol. 9, pp. 56 354–56 363, 2021.

[65] S. Su, J. She, K. Li, X. Wang, and Y. Zhou, "A nonlinear safety equilibrium spacing based model predictive control for virtually coupled train set over gradient terrains," *IEEE Transactions on Transportation Electrification*, 2021.

[66] Y. Liu, R. Liu, C. Wei, J. Xun, and T. Tang, "Distributed model predictive control strategy for constrained high-speed virtually coupled train set," *IEEE Transactions on Vehicular Technology*, 2021.

[67] S. B. Prathiba, G. Raja, K. Dev, N. Kumar, and M. Guizani, "A hybrid deep reinforcement learning for autonomous vehicles smart-platooning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13 340–13 350, 2021.

[68] A. Coppola, A. Petrillo, R. Rizzo, and S. Santini, "Adaptive cruise control for autonomous electric vehicles based on q-learning algorithm," in *2021 AEIT International Annual Conference (AEIT)*. IEEE, 2021, pp. 1–6.

[69] L. Ning, Y. Li, M. Zhou, H. Song, and H. Dong, "A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3469–3474.

[70] M. Shang, Y. Zhou, and H. Fujita, "Deep reinforcement learning with reference system to handle constraints for energy-efficient train control," *Information Sciences*, vol. 570, pp. 708–721, 2021.

[71] "X2Rail-3 Deliverable 6.1 - Virtual Train Coupling System Concept and Application Conditions," 2020. [Online]. Available: https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-3

[72] M. Schenker, R. Parise, and J. Goikoetxea, "Concept and performance analysis of virtual coupling for railway vehicles," in *Proceedings of the 3rd SmartRaCon Scien-*

tific Seminar, vol. 38. Deutsches Zentrum für Luft-und Raumfahrt eV Institut für Verkehrssystemtechnik, 2021, pp. 81–91.

[73] E. van Nunen, F. Esposto, A. K. Saberi, and J.-P. Paardekooper, "Evaluation of safety indicators for truck platooning," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1013–1018.

[74] C. Fiori, K. Ahn, and H. A. Rakha, "Power-based electric vehicle energy consumption model: Model development and validation," *Applied Energy*, vol. 168, pp. 257–268, 2016.

[75] Y. Wu, S. E. Li, J. Cortés, and K. Poolla, "Distributed sliding mode control for nonlinear heterogeneous platoon systems with positive definite topologies," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1272–1283, 2019.

[76] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[77] A. Farag, O. M. AbdelAziz, A. Hussein, and O. M. Shehata, "Reinforcement learning based approach for multi-vehicle platooning problem with nonlinear dynamic behavior."

[78] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[79] M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, and R. Ke, "Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102662, 2020.

[80] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of ids for iot: recent updates, security issues, and challenges," *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 3211–3243, 2021.

[81] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[82] "SUMO - Simulation of Urban MObility." [Online]. Available: https://www.eclipse.org/sumo/

[83] "Anylogic." [Online]. Available: https://www.anylogic.com/

[84] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[85] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *2019 IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2019, pp. 2765–2771.

[86] T. Chu and U. Kalabić, "Model-based deep reinforcement learning for cacc in mixed-autonomy vehicle platoon," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4079–4084.

[87] M. Mongelli, E. Ferrari, M. Muselli, and A. Fermi, "Performance validation of vehicle platooning through intelligible analytics," *IET Cyber-Physical Systems: Theory & Applications*, vol. 4, no. 2, pp. 120–127, 2019.

[88] A. Carlevaro and M. Mongelli, "A new svdd approach to reliable and explainable ai," *IEEE Intelligent Systems*, vol. 37, no. 2, pp. 55–68, 2021.

[89] B. Caiazzo, D. G. Lui, A. Petrillo, and S. Santini, “Distributed double-layer control for coordination of multiplatoons approaching road restriction in the presence of iov communication delays,” *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4090–4109, 2021.

[90] A. Petrillo, A. Pescape, and S. Santini, “A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks,” *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1134–1149, 2020.

[91] L. M. Castiglione, P. Falcone, A. Petrillo, S. P. Romano, and S. Santini, “Cooperative intersection crossing over 5g,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 303–317, 2020.

[92] M. Di Vaio, P. Falcone, R. Hult, A. Petrillo, A. Salvi, and S. Santini, “Design and experimental validation of a distributed interaction protocol for connected autonomous vehicles at a road intersection,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9451–9465, 2019.

[93] G. Fiengo, D. G. Lui, A. Petrillo, S. Santini, and M. Tufo, “Distributed robust pid control for leader tracking in uncertain connected ground vehicles with v2v communication delay,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1153–1165, 2019.

[94] M. Di Vaio, G. Fiengo, A. Petrillo, A. Salvi, S. Santini, and M. Tufo, “Cooperative shock waves mitigation in mixed traffic flow environment,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4339–4353, 2019.

[95] A. Petrillo, A. Salvi, S. Santini, and A. S. Valente, “Adaptive multi-agents synchronization for collaborative driving of autonomous vehicles with multiple communication delays,” *Transportation research part C: emerging technologies*, vol. 86, pp. 372–392, 2018.