

FP5TRAN S4M-R

Transforming
Europe's Rail Freight

D29.7

Execution of test for gates installed in Netherlands

Project acronym:	FP5-TRANS4M-R
Starting date:	2022-07-01
Duration (in months):	54
Call (part) identifier:	HORIZON-ER-JU-2022-01 (Topic: HORIZON-ER-JU-2022-FA5-01)
Grant agreement no:	101102009
Due date of deliverable:	Month 39
Actual submission date:	2026-04-09
Responsible/Author:	Dennis Veld (PRORAIL)
Dissemination level:	PU
Deliverable Type:	R – Document, report
Doc Version & Status:	V2.0 Submitted

Reviewed: (Yes)



"Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Europe's Rail Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them."

"The project FP5-TRANS4M-R is supported by the Europe's Rail Joint Undertaking and its members."

Document history		
<i>Revision</i>	<i>Date</i>	<i>Description</i>
V0.1	2025-11-20	First version
V0.2	2025-12-10	Review received
V1.0	2025-12-22	Final version
V2.0	2026-04-09	Revised and Resubmitted

Report contributors		
<i>Name</i>	<i>Beneficiary Short Name</i>	<i>Details of contribution</i>
Tijink, Melissa	University of Twente	Author
Veld, Dennis	ProRail	Author

Reviewers		
<i>Name</i>	<i>Beneficiary Short Name</i>	<i>Details of contribution</i>
Mónica Pelegrin	ADIF	Official Review and Feedback
ALEXIS ANDRES VIDAURRE CELIS	ADIF	Official Review and Feedback
Inigo Adin Marcos	CEIT	Official Review and Feedback

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author’s view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

The content of this report does not reflect the official opinion of the Europe’s Rail Joint Undertaking (EU-Rail JU). Responsibility for the information and views expressed in the report lies entirely with the author(s).

Table of Contents

1	Executive Summary	6
2	Abbreviations & Acronyms	7
3	Background	8
4	Objective/Aim.....	9
4.1	Task description	9
5	Content of the deliverable	10
5.1	Description of installed hardware at test site.....	11
5.2	Recognition of Hazardous Materials Plates/Placards	13
5.2.1	Goal as described in 29.4	13
5.2.2	Corresponding experiments – Hazmat number plates.	14
5.2.3	Corresponding experiments – Hazmat placards.....	18
5.3	Reading Wagon Codes.....	23
5.3.1	Goal as described in 29.4	23
5.3.2	Corresponding Experiments	23
5.4	Wagon counting	30
5.4.1	Goal as described in Deliverable 29.4	30
5.4.2	Corresponding experiments	30
6	Conclusions	32
7	References.....	35

List of Figures

Figure 1	Locations of the 5 Video Gates at Botlek.....	11
Figure 2	Example of the Video Gate at position 2	12
Figure 3	Example of the Video Gate at position 1	13
Figure 4	Example of a hazmat plate (left: the orange plate) and hazmat placards (right: the two-diamond shaped placards) – Image originates from private data collection.	14
Figure 5	Diagram of the RID/ADR number recognition pipeline. The input is a single frame/image. Following step 1, placard detection with an object detection model, the placard is extracted using bounding boxes (red box). Finally, step 2, OCR, can be applied to recognize the number. This image is copied from [R1]	14

Figure 6 Qualitative results of our proposed pipeline with YOLO11x[YOLO] and Idefics2 [IDF2]. Ground truth bounding box is shown in blue, the predicted bounding box and text are shown in purple below the plate. A) Correctly detected plate. B) Missed plate due to small plate size and the appearance of a letter instead of digits only. C) Missed plate due to low image quality. D) Correctly detected plate at night and missed plate due to skewed surface. E) Plate detected, but OCR error due to rain. A, B, and C originate from the Haztruck Dataset, D and E are from the ProRail Dataset. All images are cropped. Image is copied from [R1].	16
Figure 7 Overview of the automatic hazmat placard detection and classification pipeline. The first model detects the placard, predicting a bounding box around the located placard. The next step is placard extraction, after which the second model classifies the placard category. The final output consists of the bounding box indicating the location and the predicted category.	18
Figure 8 Hazmat placards taken from the Dutch Institute for Public Health and the Environment (RIVM) [RIVM]. Bottom: damaged placards from the DB wagon dataset. Image (a) is the symbol for corrosive materials, (b) for spontaneously combustible materials, and (c) for flammable gas. Damaged placards affect the performance of the computer vision models.	19
Figure 9 Examples of generated hazmat placards with damage	19
Figure 10 Example of a 12-digit UIC Number (33 80 783 2 199 – 3). This image comes from private data collection	23
Figure 11 Examples of images in Image Dataset. All images are provided by [STEP]	24
Figure 12 Three frames from the video dataset, illustrating the variation in code location, font style, format, and capture conditions (e.g., lighting, weather). Note that the original capture date has been cropped from each image to remove temporal metadata.	24
Figure 13 Examples of labelled challenges found in the video dataset. Only partial codes are shown.	25
Figure 14 A line-scan image.	25
Figure 15 Examples of the 5 most common challenges found in the line-scan dataset. Only partial codes are shown.	26
Figure 16 Video data pipeline evaluation. The video is evaluated frame by frame, after which a sliding window algorithm is applied to retrieve unique UIC code predictions.	27
Figure 17 Qualitative examples of UIC code predictions made by the STEP [STEP] model and our model. The ground truth is shown in green (top right), the prediction of STEP [STEP] in red (bottom left), our model in blue (bottom right), and our model with extra preprocessing in purple (top left). The checkmarks and crosses indicate if the predicted code is valid (according to Luhn's algorithm). The left images show correct UIC code predictions by our model; the right images show, from top to bottom, an image requiring extra preprocessing, a code that was missed entirely, and a false positive code in the last image. All images are cropped versions from [STEP]	28

List of Tables

Table 1 Performance metrics for YOLO11x [YOLO] and Faster RCNN [F-RCNN] under different training strategies on the two test sets. The following training strategies were used: Early Stopping (E.S), Lower Learning rate (L.L), and Data Augmentation (D.A.)	16
Table 2 Performance comparison of Tesseract [TA], EasyOCR [E-OCR], and Idefics2 [IDF2] models on ProRail and Haztruck datasets, based on the Character Error Rate (CER) and Word Error Rate (WER) in %. Lower CER and WER mean better performance.....	17
Table 3 OCR results with Tesseract [TA], EasyOCR [E-OCR], and Idefics2 [IDF2] on different cropped images. The crops are based on the ground-truth labels from the HazTruck dataset. A hyphen (-) means that the model failed to detect the code	17
Table 4 Accuracy of our proposed pipeline on the ProRail and Haztruck datasets at different IoU thresholds. The pipeline uses the YOLO11x [YOLO] and Idefics2 [IDF2] models.	17
Table 5 Distribution of Hazmat placard images by class in the DB placard set.	20
Table 6 Results on the DB wagon dataset. Highlighted values are the best-performing models in their respective Groups. All metric values are shown in percentages (%). TADS: Traditional Augmentation for Damage Simulation, GANg: GAN generated dataset, 2 SOTA dataset: HAZMAT-13 [HM-13] and DOT-HAZMAT [D-HM] Test split: 10% split from the training data, RGB: colour images of the DB-wagon dataset, B&W: Black-and-white DB-wagon dataset, IoU: Intersection over Union, and mAP : Mean Average Precision....	22
Table 7 Explanation of evaluation metrics with examples. Valid means the control digit corresponds to the code, pred = prediction, gt = ground truth (the actual UIC code).....	27
Table 8 Comparison of OCR model performance on the STEP Image Dataset [STEP], demonstrating that our proposed method achieves substantially higher accuracy, recall, and precision than existing baselines.	28
Table 9 Performance evaluation of our proposed model on video dataset subsets. Each row compares the model's performance on frames filtered by the condition in the leftmost column (subset) versus its complement (all other frames). Results show that occlusions and low-contrast conditions lead to notable drops in accuracy and recall. ...	29
Table 10 Comparison of the STEP [STEP] and our proposed model on line-scan data. Our model demonstrates substantially higher accuracy, recall, and precision across both individual camera views and combined camera views.....	30
Table 11 Performance Evaluation of our proposed model on top 5 most common challenges affecting line-scan data and their complement subsets. Each row compares the model's performance on frames filtered by the condition in the leftmost column (subset) versus its complement (all other frames). Results show that occlusions and low-contrast conditions lead to notable drops in accuracy and recall.....	30

1 Executive Summary

Deliverable (D) 29.7: “Execution of test for Gates in the Netherlands” contains the results of the tests as described in deliverable (D) 29.4: Installation and test plan for Gates in the Netherlands as made by the Netherlands (ProRail/University of Twente). The aim of the tests was to assess the Video Gates and software build in the Dutch Railroad infrastructure, and this deliverable describes the results of the test.

This document was created by experts from ProRail (Dutch Railway Company) and the University of Twente. The aim of this deliverable is to describe how the tests were conducted and to evaluate whether the test objectives regarding the physical requirements for the Video Gates, as well as software aspects, have been achieved. The test designs follow the requirements defined in Deliverable 25.1 [D25.1]. Performance metrics were defined for each requirement and the tests were conducted with those criteria in mind.

This deliverable is the result of the research done and describes the results and outcomes of the tests.

Keywords: Dutch Railroad Infrastructure, Video Gates, AI Software, Video Recognition, Quality Standards, Repeatability:

2 Abbreviations & Acronyms

Abbreviation / Acronym	Description
WP	Work package
D	Deliverable
AI	Artificial Intelligence
UT	University of Twente
HazMat	Hazardous Materials
UIC	Union internationale des chemins de fer
FR	Functional Requirements
NFR	Non-Functional Requirements
OCR	Optical Character Recognition
ILU	Intermodal Loading Unit
IoU	Intersection over Union
IVG	Intelligent Video Gate
GDPR	General Data Protection Regulation
CER	Character Error Rate
WER	Word Error Rate
GT	Ground Truth
SOTA	State-of-the-Art
Regex	Regular Expression

3 Background

The present document constitutes Deliverable (D) 29.7: “Execution of test for Gates in the Netherlands” under Flagship Project FP5- TRANS4M-R, as described in the Multi-Annual Work Plan (MAWP) for EU-RAIL. This specific activity falls under D29.2, which focuses on the deployment and validation of intelligent monitoring systems for rail operations in the Netherlands.

This deliverable outlines the installation and evaluation results for Video Gates, the AI-based software and camera systems developed by ProRail and the University of Twente. Video Gates are designed to automatically detect and assess conditions of trains and wagons as they pass through dedicated monitoring points, thereby improving safety, operational efficiency, and asset management.

This report describes how the tests were performed, the results, and the scientific papers following from this research.

4 Objective/Aim

The objective of this deliverable is to describe the results of the installation for the Video Gates developed for ProRail and the University of Twente. This report will describe the tests, as well as the algorithm used for hazmat plate, hazmat placard and UIC code detection. This report serves as the outcome for the practical testing activities carried out in D29.7 "Execution of test for gates installed in the Netherlands".

The deliverable reports on:

- The installation covering the deployment of AI software and camera hardware at designated rail monitoring locations.
- The setup and execution of tests for hazmat plate, hazmat placard and UIC code detection.

The aim is to ensure that the Video Gates can be deployed and evaluated in a systematic, measurable, and replicable manner, enabling informed validation of the technology and supporting the broader objectives of the TRANS4M-R project.

4.1 Task description

Task 29.7 started during the spring in 2025 and the creation of this deliverable started in project month 37 (July 2025). The outputs of this task are included in this document.

The Table below shows the task definition from the proposal, the output, and a link to the section where more details can be found.

	Task definition from GA (Task X.X)	Output of deliverable
Subtask 29.7	Execution of test for gates installed in the Netherlands: This task will result in an installation report and test execution for each of the considered countries (D29.2-29.7).	D29.7 Report on Execution of test for gates installed in the Netherlands

5 Content of the deliverable

This section provides the key experimental results obtained for the project's three must-have capabilities: **Hazmat detection**, **UIC code detection**, and **Wagon Counting**. The section begins with a brief description of the **hardware installed at the test site** to provide the technical context required for understanding the subsequent evaluations.

Following the hardware description, we present the test setup, evaluation methodology, data characteristics, and performance indicators associated with each must-have capability. Where relevant, complementary exploratory analyses are included, such as on the detection of damaged hazardous-material placards within the Hazmat task, or the assessment of day-night domain shift and occlusion robustness in UIC code detection.

The results summarised in this section are supported by a body of technical material comprising peer-reviewed publications [R1], publicly accessible master's and bachelor's theses [R2, R3, R4], manuscripts in preparation, and a set of internal student project reports. All thesis and student projects were supervised by one or more members of the Work Package (Melissa Tijink, Luuk Spreeuwiers, Estefania Talavera, Nicola Strisciuglio, and Shunxin Wang), ensuring alignment with the technical objectives and research direction of this deliverable.

The student project reports were produced for internal research and educational purposes and were not originally intended for public dissemination. For reasons of privacy and data protection, these reports are therefore not shared. Nevertheless, they provided valuable exploratory analyses and informed the development of subsequent research activities. The topics addressed include freight-wagon counting, comparative computer-vision approaches, graffiti detection, and preliminary experiments on UIC and container-code recognition. Readers should note that these internal documents served as supporting exploratory work rather than finalised research outputs; relevant peer-reviewed publications derived from this work are referenced where appropriate.

An overview of all conducted projects is provided below, with references to public works:

- **Hazmat Detection**

- *Vision on the Move: Automated Hazardous Material Plate Detection in Freight Transport [R1]*
- *Robust Detection of Damaged Hazardous Material Placards on Freight Vehicles [R3]*
- *Automatic Hazard Placard Detection* (student work)

- **Code Detection**

- *Optical Character Recognition in Automatic Wagon Code Reading on Freight Trains* (manuscript in preparation)
- *Wagon Number Localization Using a Vision-Language Model [R2]*

- *Freight Train Detection Technologies: A Literature Review* (student work)
- *Automatic Reading of Container Codes using VLM* (student work)
- **Wagon Counting**
 - *A Literature Review of Counting Freight Train Wagons Using Computer Vision* (student work)
 - *The Performance of Different Computer Vision Techniques in Freight Train Wagon Counting* (student work)
- **Additional Robustness Investigations**
 - *Comparative Analysis of Deep Learning Models for Graffiti Detection on Train Wagons* (student work)
 - *Enhancing Computer Vision in Adverse Weather: A Novel Model for Rain Synthesis and Removal* [R4]

Together, these materials provide the methodological and empirical basis for the results presented in this section and offer additional context for interpreting the performance and limitations of the implemented capabilities.

5.1 Description of installed hardware at test site

The Botlek Railway Yard at the Port of Rotterdam is equipped with five camera systems (see Figure 1 below) that can detect all freight trains entering and exiting the yard.

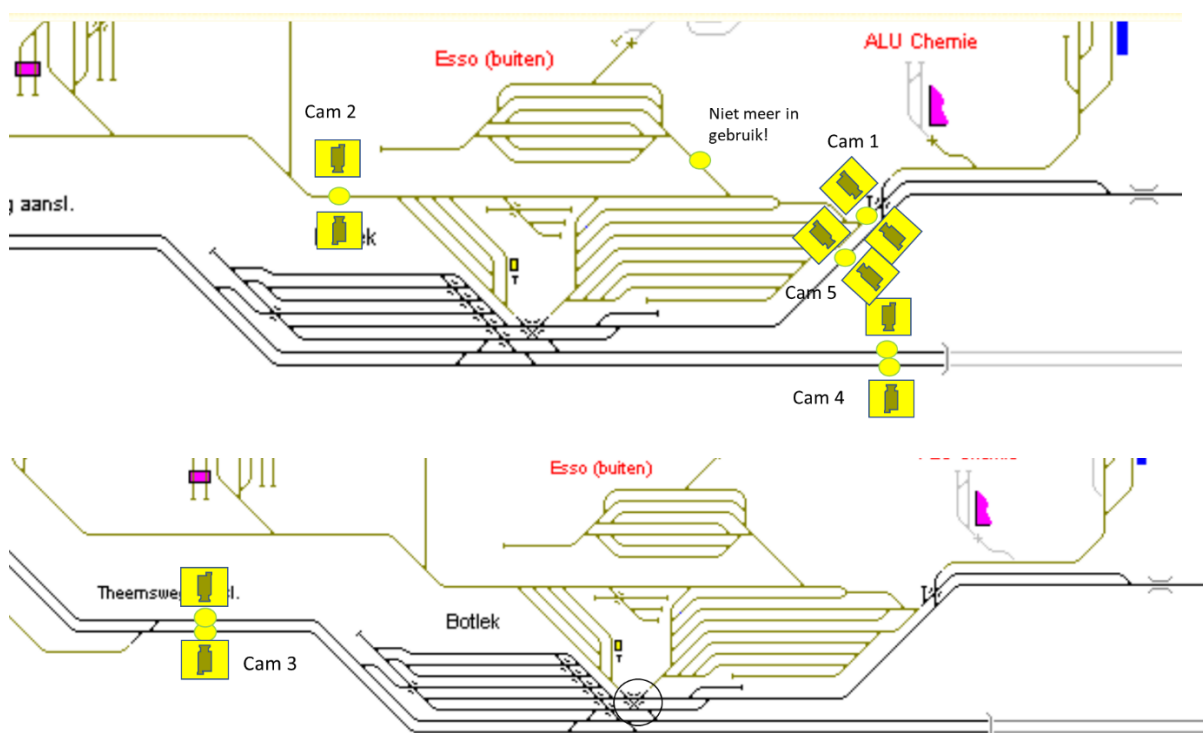


Figure 1 Locations of the 5 Video Gates at Botlek.

These systems comprise two line scan cameras (at locations 2 and 3, respectively) and three area scan cameras (at locations 1, 4 and 5, respectively).

The line scan camera can capture up to 13.000 image lines per second, which is more than enough for slow-moving trains (<10 m/s). At a train speed of 10 m/s and a camera speed of 10.000 lines/s, every image line is 1 mm wide.

The resulting image resolution is more than enough for OCR. The system can also process trains travelling at slightly higher speeds, e.g. for scan gates located at or just before the entrance of the train yard.

The current system processes colour images, for which it needs white light. Infrared light could also be used if needed, but it should be noted that not all UIC numbers show correctly in the IR domain.

More light allows for a greater depth of field and therefore sharper images. However, as too much light can blind the train driver, prudent trade-offs must be made in this regard.

The minimum light level supported by the system is 1000 lux (provided by the lighting system).

An example of a Video Gate is shown in Figure 2.



Figure 2 Example of the Video Gate at position 2

Each individual Video Gate with an area scan camera is equipped with a single 60fps Dahua IP camera and one set of Vario LED lights with motion sensors

An example of this system is shown in Figure 3.



Figure 3 Example of the Video Gate at position 1

5.2 Recognition of Hazardous Materials Plates/Placards

Trains have two types of Hazardous Materials (hazmat) signage: an orange, rectangular plate with two lines of numbers (UN-number) and diamond-shaped placards. The orange plates indicate the precise material being transported, while diamond-shaped placards indicate the general hazards associated with the material, such as flammability or oxidization. An example of both types of hazmat signage is shown in Figure 4

5.2.1 Goal as described in 29.4

Recognition of Hazardous Materials (hazmat) signage: verify if the camera/software can accurately recognize and record signage (see Figure 4) on wagons.



Figure 4 Example of a hazmat plate (left: the orange plate) and hazmat placards (right: the two-diamond shaped placards) – Image originates from private data collection.

5.2.2 Corresponding experiments – Hazmat number plates.

To automatically read the hazmat numbers, we designed a pipeline that detects the plates, subsequently extracts the plates and then subjects them automatic number reading, also known as Optical Character Reading (OCR). The pipeline is visualized in Figure 5.

We experimented with various algorithms for locating the plates and reading the numbers and selected the most effective methods for both steps. For all details, please refer to the full paper [R1].

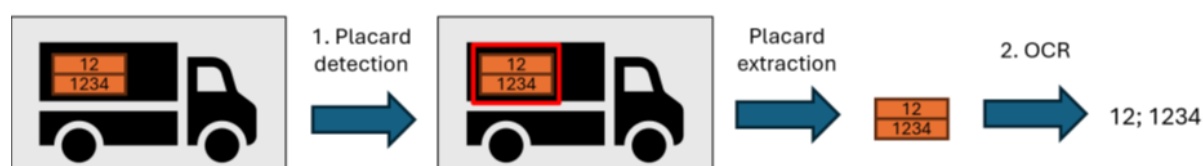


Figure 5 Diagram of the RID/ADR number recognition pipeline. The input is a single frame/image. Following step 1, placard detection with an object detection model, the placard is extracted using bounding boxes (red box). Finally, step 2, OCR, can be applied to recognize the number. This image is copied from [R1]

To train these algorithms, we created a dataset based on video data captured by a ProRail IVG. Dataset diversity is good, as video lengths vary and illumination conditions differ as a result of the time of day (day/night) and weather conditions. Only frames with a hazmat plate were considered. In total, the dataset contains 9,560 frames, extracted from 24 annotated videos, with 13 unique UN hazmat codes. The dataset was divided into training, validation, and testing subsets following an 80%-10%-10% split, resulting in 7,648 frames for training, 956 frames for validation, and 956 frames for testing. The plates were annotated using bounding boxes, and ADR/RID and UN numbers were manually

annotated. In addition to the IVG data, we also created HazTruck, a public dataset consisting mostly of images of trucks.

We evaluated the methodology in three steps: placard detection, OCR, and the full pipeline. Placard detection was evaluated using the Intersection over Union (IoU), see Equation (1), of the predicted bounding box and the labelled ground truth. The mean Average Precision (mAP) can be computed for different IoU thresholds. The mAP represents the average area under the precision-recall curve for all classes. The IoU threshold determines whether a prediction is considered correct or false. A high mAP means both good precision and recall, meaning a better model.

$$IoU = \frac{A \cap B}{A \cup B} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (1)$$

The OCR step was evaluated using the Character Error Rate (CER) and Word Error Rate (WER), shown in Equations (2) and (3), respectively. Both metrics are based on the minimum number of operations (deletions, insertions, and substitutions) needed to transform one text into another. CER measures errors at the character level, while WER evaluates accuracy at the word level. Both lower CER and WER mean better model performance.

$$CER = \frac{\text{Number of incorrect characters}}{\text{Total number of characters}} \quad (2)$$

$$WER = \frac{\text{Number of incorrect words}}{\text{Total number of words}} \quad (3)$$

The evaluation framework for the pipeline assesses performance across two key stages: detection and OCR. Placard detection is evaluated using the Intersection over Union (IoU) metric, where a detection is considered valid if its IoU with at least one ground-truth (GT) bounding box exceeds a predefined threshold. Detections are considered invalid if this threshold is not reached or if no detection is made. OCR is applied to successful detections to extract the UN numbers, which are then compared to GT values. A prediction is correct only if both numbers match the GT, while any mismatch results in an error. We evaluated overall performance by measuring accuracy, or the number of correct predictions compared to the total number of predictions. Higher accuracy means better performance of the full pipeline.

For the detection stage we considered two models: Yolo11x [YOLO] and Faster-RCNN [F-RCNN]. Details on the models can be found in the full paper. Both models are trained on detecting hazmat plates, using the annotated training portion of our labelled test set of ProRail IVG data.

The performance results of the two detection models are shown in Table 1. The details on the different training strategies can be found in the full paper. The results show little difference between the two models, although the first model (Yolo11x) shows slightly

better performance on the second dataset than HazTruck. The lower performance in general on the second dataset can be attributed to the larger variation in parameters such as distance to the camera, angle, and resolution. Since the models are trained on similar data as the ProRail Test set. We also show some qualitative results in Figure 6.

Table 1 Performance metrics for YOLO11x [YOLO] and Faster RCNN [F-RCNN] under different training strategies on the two test sets. The following training strategies were used: Early Stopping (E.S), Lower Learning rate (L.L), and Data Augmentation (D.A.)

Model	ProRail-Test		HazTruck	
	mAP50-95(%)	mAP50(%)	mAP50-95(%)	mAP50(%)
YOLO11x [YOLO]	57.69	99.37	64.16	82.43
YOLO11x E.S.	50.72	98.77	55.29	85.40
YOLO11x E.S.+L.L.	56.46	99.38	62.43	89.46
Faster-RCNN [F-RCNN]	56.34	98.92	46.1	70.57
Faster-RCNN D.A.	61.67	98.98	53.89	79.40
Faster-RCNN E.S.+D.A.	61.25	98.97	50.83	82.38

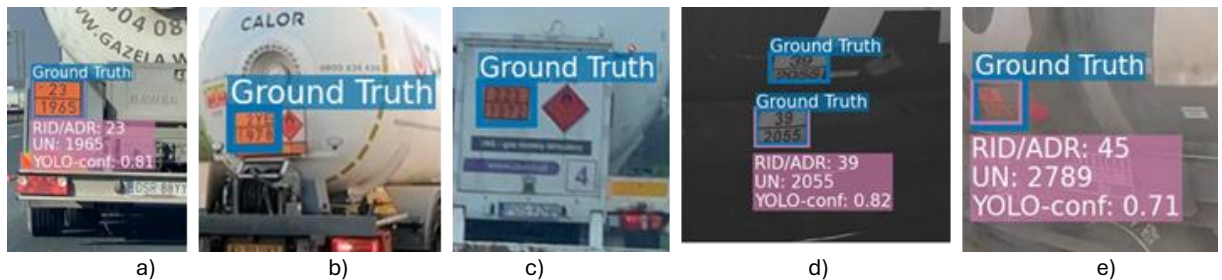


Figure 6 Qualitative results of our proposed pipeline with YOLO11x [YOLO] and Idetfics2 [IDF2]. Ground truth bounding box is shown in blue, the predicted bounding box and text are shown in purple below the plate. A) Correctly detected plate. B) Missed plate due to small plate size and the appearance of a letter instead of digits only. C) Missed plate due to low image quality. D) Correctly detected plate at night and missed plate due to skewed surface. E) Plate detected, but OCR error due to rain. A, B, and C originate from the Haztruck Dataset, D and E are from the ProRail Dataset. All images are cropped. Image is copied from [R1].

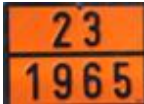
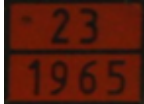
The results of the OCR step are shown in Table 3. The WER and CER scores are shown for three different models: Tesseract [TA], EasyOCR [E-OCR] and Idetfics2 [IDF2]; details on these models and their implementation can be found in the full paper [R1]. Some qualitative examples are shown in Table 2. Although the third model, Idetfics2, clearly outperforms the other models, even the best performing model can fail as a result of low-resolution images or intra-plate font variations (Table 3).

Table 2 Performance comparison of Tesseract [TA], EasyOCR [E-OCR], and Idefics2 [IDF2] models on ProRail and Haztruck datasets, based on the Character Error Rate (CER) and Word Error Rate (WER) in %. Lower CER and WER mean better performance.

Model	Prorail CER	Prorail WER	HazTruck CER	HazTruck WER
Tesseract [TA]	56.3	182.19	45.88	129.70
EasyOCR [E-OCR]	29.99	98.6	44.46	130.12
Idefics2 [IDF2]	4.85	15.20	13.28	39.74

Note: CER and WER values can actually be larger than 1, e.g. if the groundtruth for a string is "ABC" and the OCR output is "ABC12345", then the CER=5/3=166.67%.

Table 3 OCR results with Tesseract [TA], EasyOCR [E-OCR], and Idefics2 [IDF2] on different cropped images. The crops are based on the ground-truth labels from the HazTruck dataset. A hyphen (-) means that the model failed to detect the codeD.

Model	Plate 1	Plate 2	Plate 3	Plate 4	Plate 5	Plate 6
						
Tesseract [TA]	90, 3082	-, 3082	-, 11965	-, 1965	-, -	-, -
EasyOCR[E-OCR]	90, 3082	-, 130827	23, 1965	23, 4965	-, -	-, 6207
Idefics2 [IDF2]	90, 3082	90, 3082	23, 1965	23, 1965	33, -	-, 1203

The performance of the full pipeline, using the best-performing models Yolo11x [YOLO] and Idefics2 [IDF2], is summarized in Table 4, where we show the overall accuracy (correctly detected and read plates) for different IoU thresholds. This table shows that we can reach over 90% accuracy on the IVG data when reading Hazmat number plates. The failed cases are qualitatively summarized in Figure 6 above, with the main causes being low resolution, weather, or skewed surfaces.

Table 4 Accuracy of our proposed pipeline on the ProRail and Haztruck datasets at different IoU thresholds. The pipeline uses the YOLO11x [YOLO] and Idefics2 [IDF2] models.

Threshold IoU	Accuracy (%)	
	Prorail-Test	HazTruck
40 %	90.9	63.5
50 %	90.3	63.5
70 %	54.2	56.1

Overall, the test results show that proposed pipeline has clear potential when it comes to detecting and reading hazardous material number plates. The model was evaluated on images of freight trains and trucks. The results show that the Idefics2 OCR model produces the best performance, with 90.9% accuracy on freight train data and 63.5% on HazTruck data. The lower accuracy score on truck images indicates a limitation in the representation of the training data. Further research should go into training the object detection stage with a more varied dataset. Furthermore, to improve performance on both datasets, the OCR model could be fine-tuned to the task of reading hazardous materials plates.

5.2.3 Corresponding experiments – Hazmat placards

There are 9 main categories of hazmat placards, with several distinguishable subclasses [RIVM]. The placards are always diamond-shaped, but multiple different placards can appear on a wagon, and the colours and pictograms on the placards can vary.

To automatically read hazmat placards, we designed a pipeline that first detects the placard and subsequently classifies it. The final output shows the original image, with a bounding box indicating the location of the placard, and a corresponding class prediction. A diagram of the full pipeline is shown in Figure 7.



Figure 7 Overview of the automatic hazmat placard detection and classification pipeline. The first model detects the placard, predicting a bounding box around the located placard. The next step is placard extraction, after which the second model classifies the placard category. The final output consists of the bounding box indicating the location and the predicted category.

The performance of the pipeline is evaluated on a DB Cargo dataset consisting of images of hazmat placard cut-outs taken from line-scan images of an IVG. As the examples in Figure 8 shows, the images in this dataset are not “perfect”: they may be scratched, torn or captured in suboptimal lighting conditions. This affects model performance.

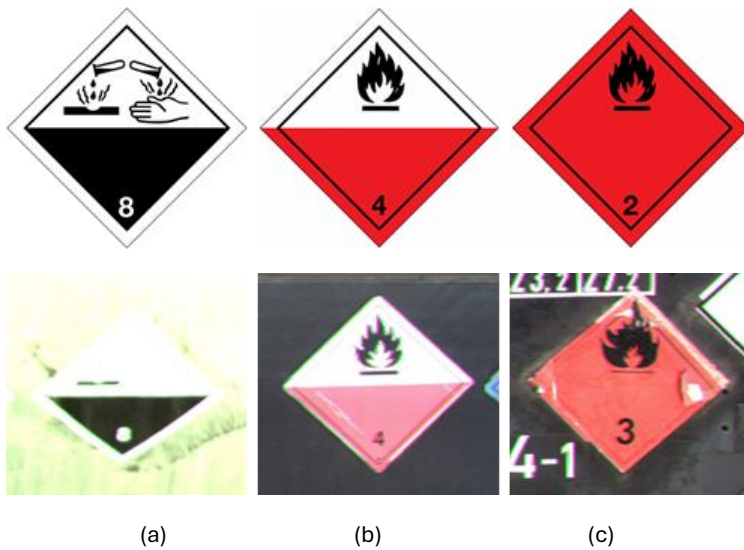


Figure 8 Hazmat placards taken from the Dutch Institute for Public Health and the Environment (RIVM) [RIVM]. Bottom: damaged placards from the DB wagon dataset. Image (a) is the symbol for corrosive materials, (b) for spontaneously combustible materials, and (c) for flammable gas. Damaged placards affect the performance of the computer vision models.

To compensate for these real-life effects, we trained the models on partially synthesized data to mimic damage such as scratches and tears. Figure 9 shows examples of these generated placards.



Figure 9 Examples of generated hazmat placards with damage

The model was evaluated using the DB placard set, as mentioned before. An overview of classes and images is shown in Table 5. Three classes that did not appear in the public data used for training the models were removed: toxic gas, miscellaneous and miscellaneous fish. Furthermore, the table also shows that not all classes are present in the DB placard set, with infectious substance and inhalation hazardous missing.

Table 5 Distribution of Hazmat placard images by class in the DB placard set.

Class	Image Count
Corrosive	120
Dangerous when wet	59
Explosive	50
Flammable	300
Flammable solid	53
Infectious substance	0
Inhalation hazardous	0
Non-flammable gas	54
Organic peroxide	50
Oxidizer	129
Poison	104
Radioactive	50
Spontaneously combustible	72
Total	1141

The results are presented using the precision, recall and F1 score, shown in Equations (4), (5) and (6). We also use Intersection over Union and Mean Average Precision, as described in the previous experiment.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (4)$$

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (5)$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

Precision shows the ratio of correctly predicted (True positives) hazmat placards to all predicted hazmat placards as a class, summarized for all classes. Recall shows the ratio of correctly predicted hazmat placards of a class to all placards that should have been predicted as that class. The F1 score is preferred over accuracy, since accuracy is not always a fair metric for imbalanced datasets with unequal numbers of images per class. The F1 score shows how sensitive the method is to errors, both false positives and false negatives.

We evaluated using different training strategies, mainly varying the training data by adding synthesized data.

Training datasets were constructed from two labelled, easily available public datasets of US placards, which contain text, unlike European placards. Both datasets were subjected to basic augmentation, called TADS in the results table. For the third experiment, we generated damaged placards meeting US and European specifications, respectively (Figure 9). In the last test, we combined all training data options. In addition to the standard full-colour DB placard dataset, we also created a black-and-white version for evaluation, as some IVGs might only collect black-and-white images. Lastly, we also evaluated the performance on the same type of data as used in the training data, which is called the “Test split” in the table. The results are presented in Table 6.

The results show that we can reach an F1 score of 86.51% on the full-colour DB placard dataset. As this is slightly lower than the maximum test split performance (Max. F1 score = 96.38%), there is still a gap between training and the DB data. The training data must therefore not yet fully cover certain situations, or damage types. Furthermore, performance was strikingly lower on black-and-white images than on full-colour, which can be explained by the fact that the models were trained on full-colour images only and that colour is an important placard feature. This discrepancy suggests that the models factor in placard colour in detection and classification, and training data should therefore also include black-and-white images to improve performance. This will also help to deal with varying lighting conditions, e.g. during twilight. The models were not evaluated on full-frame IVG images, as this requires manual labelling of the IVG data, which has not been done yet. Overall, the models do show great potential for correctly identifying hazmat placards, especially considering the challenging real-world conditions, such as damage, in the DB dataset.

Table 6 Results on the DB wagon dataset. Highlighted values are the best-performing models in their respective Groups. All metric values are shown in percentages (%). TADS: Traditional Augmentation for Damage Simulation, GANg: GAN generated dataset, 2 SOTA dataset: HAZMAT-13 [HM-13] and DOT-HAZMAT [D-HM] Test split: 10% split from the training data, RGB: colour images of the DB-wagon dataset, B&W: Black-and-white DB-wagon dataset, IoU: Intersection over Union, and mAP : Mean Average Precision.

Model + training data	Dataset	Precision	Recall	F1 Score	IOU	mAP@0.5	mAP@0.5:0.95
Models trained on 2 SOTA	Test split	95.29	96.34	95.81	93.81	92.93	82.25
	DB wagon (RGB)	82.33	76.56	79.34	93.12	64.94	56.22
	DB wagon (B&W)	45.32	38.14	41.42	92.18	23.62	20.28
Models trained on 2 SOTA + TADS	Test Split	96.35	95.43	95.89	93.63	92.51	81.78
	DB wagon (RGB)	86.21	75.70	80.61	94.38	65.99	59.49
	DB wagon (B&W)	58.86	47.55	52.60	93.32	35.44	30.78
Models trained on 2 SOTA on GANg (with US signs) data	Test split	95.67	96.68	96.17	93.84	93.51	82.64
	DB wagon (RGB)	70.76	62.54	66.39	93.26	47.59	41.36
	DB wagon (B&W)	38.81	28.15	32.63	92.87	11.71	9.93
Models trained on 2 SOTA + TADS + GANg (with US signs) data	Test split	96.26	95.34	95.80	93.54	92.09	80.52
	DB wagon (RGB)	88.10	77.52	82.47	94.37	68.73	61.86
	DB wagon (B&W)	54.41	44.48	48.94	93.31	30.94	26.84
Models trained on 2 SOTA + GANg (with EU and US signs) data	Test Split	95.85	96.92	96.38	93.90	93.45	83.02
	DB wagon (RGB)	81.28	73.39	77.13	93.20	62.75	54.41
	DB wagon (B&W)	39.41	30.74	34.54	91.89	14.56	12.49
Models trained on 2 SOTA + TADS + GANg (with EU and US signs) data	Test split	96.17	95.24	95.70	93.56	92.37	81.32
	DB wagon (RGB)	91.99	81.65	86.51	94.56	75.98	68.92
	DB wagon (B&W)	57.21	46.49	51.30	93.33	26.77	23.33

Compared to the requirements formulated in D25.1 (identify dangerous goods signs on containers on wagons passing the camera on the track and the system detection rate reaches 95%), we show that on the test split this goal has been reached (precision and recall both >95%), however, on the DB wagon data the performance is somewhat lower (recall=92% and [precision=82%). We expect this can be improved by including DB wagon data in the training set.

5.3 Reading Wagon Codes

Wagon numbers, or UIC numbers, are 12-digit numerical codes that uniquely identify a wagon, as shown in Figure 10. Automatically reading these codes from the IVG data would make it possible to automatically create a wagon list and verify said list along the whole route.

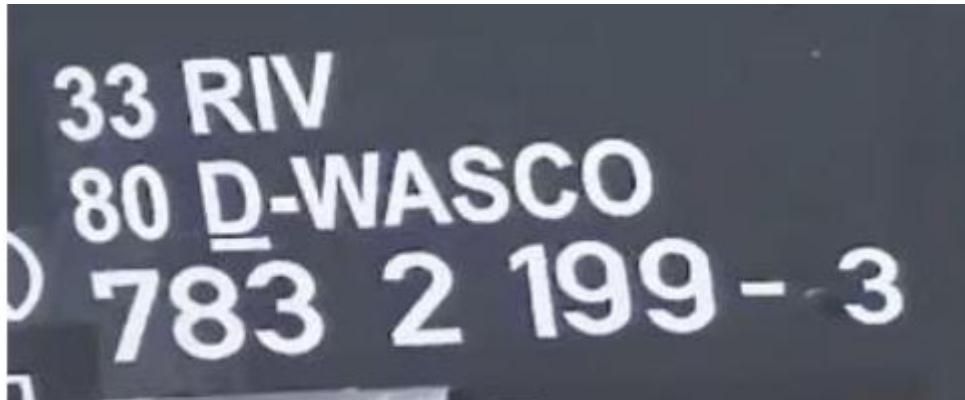


Figure 10 Example of a 12-digit UIC Number (33 80 783 2 199 - 3). This image comes from private data collection

5.3.1 Goal as described in 29.4

Ensure the system can identify and log UIC numbers (see Figure 10) on passing wagons.

5.3.2 Corresponding Experiments

To automatically read UIC numbers, we designed a pipeline that detects and reads UIC codes from an input image or video frame. The pipeline is based on the STEP method, designed by [STEP]. Uniquely, this method performs localization and reading in the same step, which other methods typically do in two steps (as in the hazmat plate method described above). To guide the search, we do not consider all text on the wagon, but help the algorithm by defining the structure of the text we are looking for. Using a regular expression (regex), we can explicitly tell the model to look only for sequences such as digit-digit-space-digit-digit-space-digit-digit-digit. A drawback of the existing STEP algorithm was its suitability for one-line text only, while UIC codes often appear on multiple lines (see Figure 10). Furthermore, code structures vary widely in reality, as they may span a large surface and may not appear in full in one easily demarcated space. By adding several modules, tapping into prior knowledge of the codes, we implement hierarchical regex matching to deal with the semi-structured nature of the UIC codes and multi-line UIC codes, instead of focusing solely on full 12-digit codes. A full description of the methodology is provided in a manuscript currently under preparation and will be published upon acceptance. Until then, any questions regarding the methodological details may be addressed to the authors of this document.

The method was evaluated on three different data types: image data, video data and line-scan data. Since several camera types are still used within the project, we preferred a

method that could handle all inputs rather than only one data type. We will first describe the datasets, then the evaluation metrics, and finally show the performance.

5.3.2.1 Image data

The dataset is provided by the authors of [STEP] and consists of 396 pictures of stationary wagons taken different distances and angles. The wagon codes have different fonts and regular expressions, but most are single-line codes. Some examples are shown in Figure 11 below:



Figure 11 Examples of images in Image Dataset. All images are provided by [STEP].

5.3.2.2 Video data

This dataset consists of video data captured by an IVG in the Netherlands over the course of several months at one location. The camera was triggered by a passing train and collected up to 10 minutes of data. Several examples of frames are shown in Figure 12. A selection of 70 videos of varying lengths and capture times was made to label the UIC codes. Codes were labelled using the interpolation tool of the Computer Vision Annotation Tool (CVAT). During the labelling process, any anticipated challenges with the codes were also noted, such as occlusions and low contrast. The full videos were also annotated, distinguishing between day/night and the presence of rain. Several examples are shown in Figure 13. In total, the dataset contains 554 unique UIC codes.



Figure 12 Three frames from the video dataset, illustrating the variation in code location, font style, format, and capture conditions (e.g., lighting, weather). Note that the original capture date has been cropped from each image to remove temporal metadata.

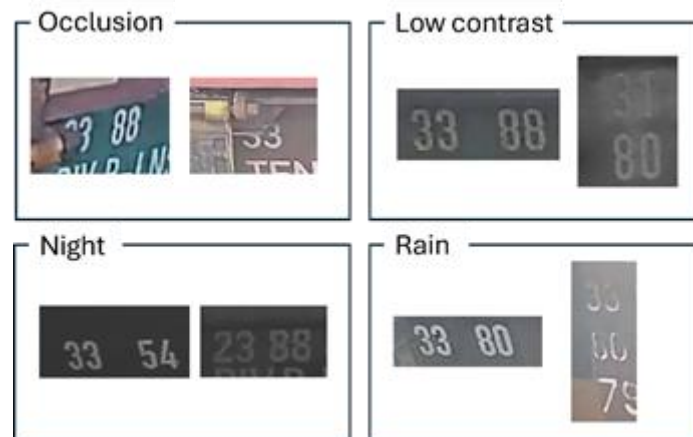


Figure 13 Examples of labelled challenges found in the video dataset. Only partial codes are shown.

5.3.2.3 Line-scan Data

The third dataset is also private and consists of grayscale line-scan images of freight wagons, as shown in Figure 14, captured by the same cameras as the data used by [W&CCDR]. There are two line-scan cameras on both sides of the train tracks, meaning that each wagon is photographed from two sides. Due to asymmetry and failed photos, the images do not fully overlap, resulting in 831 images total, of which 430 are from one camera and 401 from the other. This dataset was also labelled using the CVAT tool. In addition to the UIC codes, challenging conditions were also noted; examples are shown in Figure 15.



Figure 14 A line-scan image.



Figure 15 Examples of the 5 most common challenges found in the line-scan dataset. Only partial codes are shown.

5.3.2.4 Evaluation metrics

To evaluate UIC code-reading performance, we have to define when the model output is correct and when not. Our definition, with examples, is shown in Table 7. The first criterion concerns the final digit of the 12-digit code, which is a control digit that serves to identify any mistakes in the prior 11 digits. The control digit is determined using Luhn's algorithm. The second criterion consists of verifying whether the predicted code matches the ground truth: the actual UIC code present in the image. Using the True/False Positives and True/False negatives, we can calculate Accuracy, Precision and Recall. The image dataset contains no True Negatives, as all images feature a UIC code.

The video frames were evaluated using a sliding window format, as combining information from several frames may produce stronger predictions. As we only need to identify unique wagons to make a wagon list, we do not consider individual frame predictions and only output unique predictions per video. The sliding window has two parameters: window length (how many frames are considered together) and a threshold (to determine the minimum number of frames within a window in which the same code has to be found to make a final prediction). Figure 16 shows a diagram of the sliding window. This approach means that we only compare the predicted list of unique UIC codes to the GT list of UIC codes, ruling out any True Negatives.

As mentioned before, two-side views are available for most line-scan images. In addition to considering all individual images, we also combine the frames. We use the validity of the code as a selection criterion to determine the final prediction for each wagon. For

example, if no code or an invalid code is found on the left-side view, while a valid code is found on the right-side view, the prediction of the second view is considered final. In this dataset True Negatives are possible, as some images do not have a UIC code, and this should also be predicted.

Table 7 Explanation of evaluation metrics with examples. Valid means the control digit corresponds to the code, pred = prediction, gt = ground truth (the actual UIC code).

METRIC	CONDITION	EXAMPLE PREDICTION (PRED)	EXAMPLE GROUND TRUTH (GT)
TP (TRUE POSITIVE)	Valid and pred = gt	31 80 0691 235 – 2	31 80 0691 235 – 2
FP (FALSE POSITIVE)	Valid and pred ≠ gt	33 85 4961 643 – 5	31 80 0691 235 – 2
		31 80 0691 235 – 2	No code
FN (FALSE NEGATIVE)	Not valid and pred ≠ gt	81 80 0691 235 – 2	31 80 0691 235 – 2
		No code	31 80 0691 235 – 2
TN (TRUE NEGATIVE)	Not valid and pred = gt	No code	No code

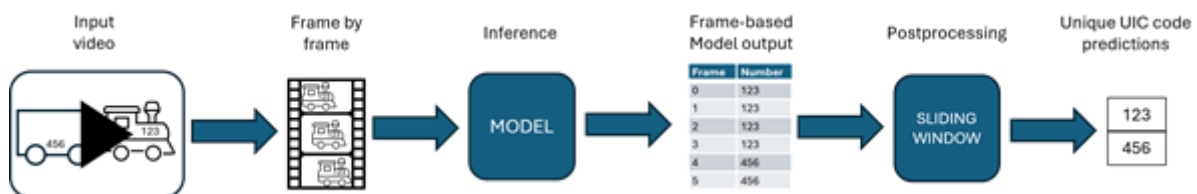


Figure 16 Video data pipeline evaluation. The video is evaluated frame by frame, after which a sliding window algorithm is applied to retrieve unique UIC code predictions.

5.3.2.5 Results

Experiment 1 – Image data

In this experiment, we evaluated the images in Dataset 1 – Image data. The results show that we can reach an accuracy of more than 79% on this data, marking a major improvement compared to the original model. There is, however, still room for improvement. Several qualitative results show examples of where the models succeeded and failed, shown in Figure 17. The results demonstrate that the main causes of incorrect predictions were difficult fonts (e.g. open fonts), occlusions, and other 12-digits codes.

Table 8 Comparison of OCR model performance on the STEP Image Dataset [STEP], demonstrating that our proposed method achieves substantially higher accuracy, recall, and precision than existing baselines.

Model	Accuracy (%)	Recall (%)	Precision (%)
STEP [STEP]	22.5	23.1	89.0
Our model	79.3	84.4	92.9

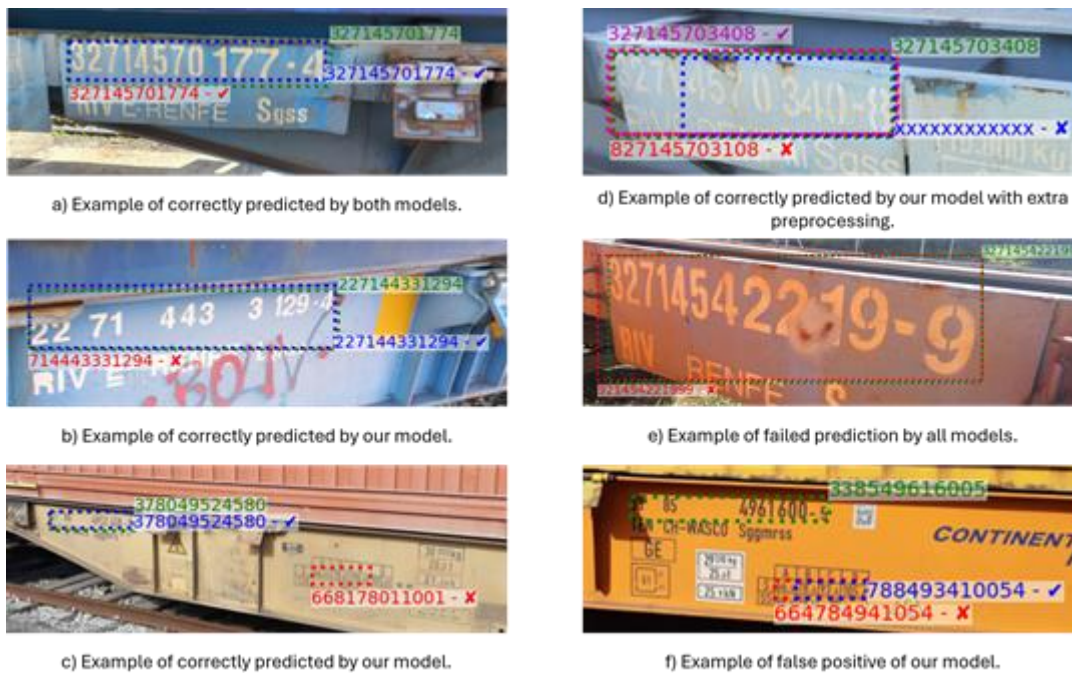


Figure 17 Qualitative examples of UIC code predictions made by the STEP [STEP] model and our model. The ground truth is shown in green (top right), the prediction of STEP [STEP] in red (bottom left), our model in blue (bottom right), and our model with extra preprocessing in purple (top left). The checkmarks and crosses indicate if the predicted code is valid (according to Luhn's algorithm). The left images show correct UIC code predictions by our model; the right images show, from top to bottom, an image requiring extra preprocessing, a code that was missed entirely, and a false positive code in the last image. All images are cropped versions from [STEP].

Experiment 2 – Video Data

Table 9 shows the results per category with a window length of 12 and a threshold of 60%. The first row shows the results on the complete dataset, with accuracy peaking at 75%. The second row shows the results for images without any reported issues, with accuracy reaching 79% for the no-issue set and 70% for the complement subset (all data with issues). For UIC codes captured at night, recall is significantly lower, indicating that this condition is more likely to cause false negatives (wrong or missed codes). We also see that occlusions and low contrast hamper model performance, as evidenced by a clear drop in performance. The combination of nighttime conditions and occlusion in particular result in very poor performance, with an accuracy score of 35.7%. The models are especially

sensitive to these errors because their core underlying principle is finding regular expressions. It is an “all-or-nothing” approach, meaning models will look only for full codes or nothing. If a part of a code is occluded, the complete code will not be found. The remaining failures (since accuracy was only 79.1% at best), could be caused by unlabelled issues, missing regular expressions, or wrongly detected codes (see Figure 17– F).

The results suggest that rain does not negatively impact performance, with the proviso that rain is only present in a small portion of the data (3.5%). While we can therefore hypothesize that it will not cause problems, future research is needed to analyse the full impact of rain on the results, focusing particularly on heavy rain and the combination of rain and nighttime/low-light conditions. This calls for a larger dataset, either collected by an IVG or synthesized by implementing synthetic rain on the available data. The latter method would provide full control over the severity of the rain and light conditions, but real data should always be used to verify the impact on real-world implementation.

Table 9 Performance evaluation of our proposed model on video dataset subsets. Each row compares the model's performance on frames filtered by the condition in the leftmost column (subset) versus its complement (all other frames). Results show that occlusions and low-contrast conditions lead to notable drops in accuracy and recall.

Subset	Percentage dataset	Metrics Subset			Metrics Complement Subset		
		Accuracy(%)	Precision(%)	Recall(%)	Accuracy(%)	Precision(%)	Recall(%)
All data	100	75.0	92.1	80.1	-	-	-
No issues	52.5	79.1	88.4	88.1	70.4	97.1	72.0
Night	28.9	71.3	96.1	73.5	76.5	90.7	83.0
Occlusion (all)	14.5	55.8	100.0	55.8	78.3	91.2	84.6
Rain	3.5	76.2	94.1	80.0	75.0	92.0	80.1
Low contrast	2.7	56.3	56.3	100.0	75.5	92.0	81.0
Night + occlusion	2.4	35.7	100.0	35.7	76.0	92.0	81.3
Occlusion only	11.7	62.3	100.0	62.3	76.7	91.3	82.7

Experiment 3 – Line-scan data

The results on the line-scan data are shown in Table 10. The model boasts 79.8% accuracy and even reaches an accuracy score of 86.2% on wagons captured by both cameras.. Combining information from different views clearly improves performance and makes the system more robust against missing codes or failed detections. Similar to the experiment on video data, we ran an extensive analysis on possible challenges within the data. The results of this analysis, focusing only individual images (not combined views), are shown in Table 11. The results show that occlusions and low contrast are particularly challenging for the model, as was also the case for the video data. Since the recall is lower in these situations, false negatives (missing codes) are more common. The last row of Table 11 shows that the accuracy is 84.7% on the subset without any labelled issues. The remaining errors are caused to mislabelled codes, missed challenges while labelling, other 12-digit codes on wagons resulting in false positives, and regular expressions missing from the algorithm's regex list. The results help pinpoint several ways to improve the overall methodology: robustness against occlusions, low contrast, and font variations. It is also

strongly advised that wagon owners address these issues by adopting more legible and standardized wagon codes.

Table 10 Comparison of the STEP [STEP] and our proposed model on line-scan data. Our model demonstrates substantially higher accuracy, recall, and precision across both individual camera views and combined camera views.

Model	Data Subset	Total	Accuracy (%)	Recall (%)	Precision (%)
STEP [STEP]	All	831	12.6	2.84	32.8
	Camera 1	430	13.5	4.08	44.1
	Camera 2	401	11.7	1.48	18.5
	Combined	507	19.1	3.84	30.6
Our model	All	831	79.8	84.6	90.4
	Camera 1	430	80.2	84.2	91.6
	Camera 2	401	79.3	85.0	89.1
	Combined	507	86.2	91.8	90.4

Table 11 Performance Evaluation of our proposed model on top 5 most common challenges affecting line-scan data and their complement subsets. Each row compares the model's performance on frames filtered by the condition in the leftmost column (subset) versus its complement (all other frames). Results show that occlusions and low-contrast conditions lead to notable drops in accuracy and recall.

Subset	Percentage dataset	Metrics Subset			Metrics Complement Subset		
		Accuracy(%)	Precision(%)	Recall(%)	Accuracy(%)	Precision(%)	Recall(%)
Font size variation	12.39	74.8	81.1	90.6	80.5	92.0	83.7
Faded	8.66	68.1	90.6	72.7	80.6	90.3	85.8
Occlusion	7.70	65.6	85.7	73.7	81.0	90.7	85.6
Low contrast	5.42	62.2	93.3	65.1	80.8	90.2	85.9
Angle	5.29	97.8	97.8	100.0	78.8	89.8	83.5
No issues	44.89	84.7	93.0	90.5	75.3	86.5	78.5

5.4 Wagon counting

The IVG should be able to count how many wagons trains have, e.g. to determine train length or verify the wagon list.

5.4.1 Goal as described in Deliverable 29.4

To confirm that the system can accurately count the number of wagons in a train.

5.4.2 Corresponding experiments

Preliminary experiments and a literature study have been conducted as part of the student projects; however, we ran no specific experiments related to this goal. Automated wagon counting was not explored for several reasons, but primarily because the chosen data type (video or line scan) would determine what method is most suitable and no final choice has been made as of yet.

Line-scan systems feature a secondary sensor that measures the speed of the train in order to determine the capture speed of the camera. This sensor might be an axle

counter, which could also be used to count wagons. Moreover, a line-scan camera outputs single-wagon images, as shown in Figure 14, and therefore automatically counts wagons too. Since we did not have direct access to the raw sensor data, we could not verify if the wagon count was correct.

For video data, counting wagons constitutes a distinct computer-vision challenge. We focused on reading UIC codes and hazmat placards, so no additional methodologies were designed for wagon counting. However, as the UIC code methodology outputs a list of unique wagon codes, the length of this list should correspond to the wagon count, which is reflected in the performance metrics of Table 9.

This means that the technology to perform wagon counting is present, however, no evaluation of the correctness of wagon counting has been conducted. This is left for a future study, when we have access to all raw sensor data from the video gate.

6 Conclusions

Deliverable D29.7 is part of the TRANS4M-R Flagship Project and presents the execution and outcomes of the tests performed on Dutch Video Gates, AI-based monitoring systems that automatically analyse passing freight trains developed jointly by ProRail and the University of Twente. In the broader project context, these Video Gates are intended to enhance safety, operational efficiency, and asset management by automating the detection of essential wagon information. The problem addressed in this work is that key safety-critical identifiers, such as hazmat signage and UIC wagon numbers, are difficult to read automatically under real-world operating conditions, especially when affected by motion, weather, low contrast, or physical degradation. The objective of this deliverable is therefore to document how the tests were conducted and to evaluate whether the installed hardware and software meet the functional and technical requirements defined in earlier deliverables, particularly D25.1 and D29.4. We assessed the performance of the computer-vision algorithms, the behaviour of the installed systems, and the extent to which the Video Gates fulfil their intended role within the Dutch railway monitoring infrastructure.

This work evaluated a set of computer-vision pipelines designed to automatically detect and read hazardous materials signage and UIC wagon numbers, and to support wagon counting. The methodology combined object detection, OCR, and structured text-recognition techniques. For hazmat plates, a two-stage pipeline was used (plate detection followed by OCR), while hazmat placards were processed by detection and classification models trained on both real and synthetically damaged data. UIC numbers were addressed with an adapted method based on regular expressions.

Focusing on the ProRail IVG data, the findings show that hazmat number plate recognition performs very well in real-life operating conditions: the full pipeline achieves over 90% accuracy on the ProRail test set when reading UN numbers. For hazmat placards, the main conclusion is that including synthetically damaged training data significantly improves real-world performance. Models trained only on publicly available datasets (mainly U.S. placards) perform notably worse on the DB wagon placard dataset, whereas training strategies that add EU-style synthetic damage produce the highest F1 scores. This clearly highlights a gap between public, clean training data and the worn, scratched, or partially degraded placards found in the real world, and demonstrates the value of synthetic augmentation for bridging this gap. Similarly, UIC recognition reaches 75% accuracy on ProRail video data and 79–86% on line-scan data, while combining information from both camera views improves performance. These results indicate that the proposed pipelines are already robust to typical IVG conditions such as speed variation, day/night settings, and minor visibility challenges.

For hazmat plates, the evaluation was carried out on a relatively small real-world test set, meaning that important sources of variability (rain, snow, or other challenging outdoor conditions) were not fully represented. For placard classification, only cropped placard images were used rather than full-frame wagon images, so the detection stage (locating the placard on the wagon) was not fully assessed, nor were systematic outdoor effects such as weather, lighting changes, or surface damage beyond what was synthetically generated. For UIC recognition on both line-scan and video data, only limited weather effects were included in the datasets, and the results show that occlusions and low-contrast situations remain the dominant causes of failure, indicating that the system still lacks robustness in these common real-world scenarios.

In addition, all evaluations were performed in an offline setting, meaning the algorithms were tested separately from the operational IVG hardware rather than in a fully integrated, real-time system. Although real-world IVG data was used throughout, the processing itself ran on external hardware, and the full end-to-end pipeline was not implemented. As a result, processing time and system efficiency were not evaluated. Assessing real-time performance on the actual hardware platform is therefore an essential next step, as the focus of this work was on demonstrating the recognition potential of the algorithms rather than their execution speed or integration readiness.

Across all evaluated components, the most effective solution combines the YOLO11x [YOLO] detector with the Idefics2 [IDF2] OCR model for hazmat number plates, delivering over 90% accuracy on ProRail data and showing strong robustness under typical IVG conditions. For hazmat placards, the classification models trained with synthetically damaged EU and US placards achieve the highest F1 scores, demonstrating that realistic augmentation is essential for real-world performance. For UIC codes, the method with hierarchical regex matching provides a substantial improvement over existing baselines across image, video, and line-scan data. Together, these components currently form the strongest-performing configuration and constitute the most reliable foundation for an integrated wagon-inspection system.

Future work should focus on moving the system towards full operational integration. A key next step is to implement wagon counting, which requires full access to all raw sensor data so that information from multiple sensors can be combined effectively. More extensive evaluation under real-world conditions, including varied weather, lighting, and environmental effects, is essential to ensure robustness. For hazmat placards, the recognition pipeline should be tested on full-frame wagon images, which will require additional labelled data to evaluate the complete detection-classification chain. Beyond recognition accuracy, the efficiency and processing time of the full pipeline must be assessed on the actual hardware to verify real-time feasibility. In the longer term,

improving resilience to occlusions and unreadable UIC codes, for example by introducing a dedicated “code not readable” flag, will further enhance reliability. Ultimately, the goal is to combine all developed algorithms into a single integrated system capable of handling plate reading, placard classification, UIC recognition, and wagon counting within one unified framework.

Overall, the deliverable successfully meets the primary objectives defined in D29.4 and D25.1, demonstrating that the installed Video Gates and associated software perform reliably in recognising hazmat plates, hazmat placards, and UIC wagon numbers under typical operating conditions. The algorithms achieved high accuracy scores on ProRail IVG data, validating their suitability for automated monitoring and confirming that the core functional goals (accurate detection, reading, and classification) have been achieved. While several limitations remain, particularly regarding evaluation under adverse weather conditions, full-frame placard detection, and real-time hardware integration, these do not undermine the conclusion that the planned objectives of this task have been achieved. The work provides clear evidence of technical viability, establishes a solid basis for further optimization, and supports the continued deployment and validation of intelligent monitoring systems within the TRANS4M-R framework.

7 References

- [R1] Tijink, M., Levendeev, S., Nieuwenhuis, E., Spreeuwiers, L., Strisciuglio, N., & Talavera, E. (2026). *Vision on the Move: Automated Hazardous Material Plate Detection in Freight Transport*. In M. Castrillón-Santana, C. M. Travieso-González, O. Deniz Suarez, D. Freire-Obregón, D. Hernández-Sosa, & J. Lorenzo-Navarro (Eds.), *Computer Analysis of Images and Patterns* (pp. 259–270). Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-04968-1_22
- [R2] Mitev, I. (2025). *Wagon Number Localization Using a Vision-Language Model* (Master's thesis). University of Twente, Enschede. Retrieved from <https://purl.utwente.nl/essays/105067>
- [R3] Kishore Kumar, P. (2025). *Robust Detection of Damaged Hazardous Material Placards on Freight Vehicles* (Master's thesis). University of Twente, Enschede. Retrieved from <https://purl.utwente.nl/essays/108395>
- [R4] Tolstobrov, K. (2025). *Enhancing Computer Vision in Adverse Weather: A Novel Model for Rain Synthesis and Removal* (Master's thesis). University of Twente, Enschede. Retrieved from <https://purl.utwente.nl/essays/108492>
- [Yolo] Jocher Glenn: Ultralytics Yolo 11, <https://github.com/ultralytics/ultralytics>, Accessed: June 2025
- [F-RCNN] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6), 1137–1149 (6 2015)
- [IDF2] Laurençon, H., Tronchon, L., Cord, M., Sanh, V.: What matters when building vision-language models? (2024). <https://doi.org/10.48550/arXiv.2405.02246>
- [TA] Hoffstaetter, S., Lee, M.: Pytesseract · PyPI, <https://pypi.org/project/pytesseract/>, Accessed: June 2025
- [E-OCR] Kittinaradorn, R.: Easyocr · PyPI, <https://pypi.org/project/easyocr/>, Accessed: June 2025
- [RIVM] Rijksinstituut voor Volksgezondheid en Milieu (RIVM). Gevaarsetiketten en kenmerken (n.d.). Retrieved December 1, 2025, from <https://rvs.rivm.nl/onderwerpen/gevaarsindeling/ADR/gevaarsetiketten-en-kenmerken>
- [HM-13] A. Sharifi, A. Zibaei, and M. Rezaei, "A deep learning based hazardous materials (HAZMAT) sign detection robot with restricted computational resources," *Machine Learning with Applications*, vol. 6, p. 100104, Dec. 2021. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2666827021000529>

[D-HM] M. Pamnani, T. Gala, S. Walawalkar, K. Jagasia, and R. Hande, "DOT-HAZMAT: An Android Application for Detection of Hazardous Materials (HAZMAT) Signs," SSRN Electronic Journal, 2022. Available: <https://www.ssrn.com/abstract=4111847>

[STEP] S. Garcia-Bordils, D. Karatzas, and M. Rusinol, "STEP – Towards Structured Scene-Text Spotting," Sep. 2023. Available: <http://arxiv.org/abs/2309.02356>.

[W&CCDR] A. Diaz-Diaz, F. Parrilla, R. D. La Iglesia, R. Barea, and L. M. Bergasa, "Wagon and Container Codes Detection and Recognition based on YOLOv8," 2024 7th Iberian Robotics Conference (ROBOT), pp. 1–6, Nov. 2024. DOI: 10.1109/ROBOT61475.2024.10796925. Available: <https://ieeexplore.ieee.org/document/10796925/>.