Wagon and Container Codes Detection and Recognition based on YOLOv8

Alejandro Diaz-Diaz* Francisco Parrilla* R. De La Iglesia* Rafael Barea† Luis M. Bergasa†

***Abstract*—Railway systems have become an essential and growing element for freight transport sector due to the low levels of pollution it offers compared to other alternatives. However, they have not seen a corresponding advancement in technology, as many of the inspection procedures are still based on outdated technologies or even performed manually. This work proposes a method for automating the reading of UIC (wagons) and BIC/ILU (containers) codes based on YOLOv8, both for detection and recognition processes. The challenge is that, unlike in the case of car license plates, codes on freight trains can be located in different parts of the train or even appear fragmented on different parts of the container, present irregular lighting conditions, or have different formats due to the lack of strict standardization. The proposed method has been compared using real images obtained from a railway operator, achieving significantly better results compared to previous works.**

***Index Terms*—ivg, yolov8, railway, code, ocr**

I. Introduction

Climate change and greenhouse gases are increasingly becoming a more significant threat worldwide. Aiming to address this issue progressively, a series of proposals and goals have been compiled in the European Green Deal [1], which seeks to slow down climate change by reducing net greenhouse gas emissions by at least 55% by 2030 and completely eliminating them by 2050.

Shifting freight transport to rail to reduce road transport has been identified as one of the potential solutions [2]. A key element for this dynamization is the digitalization of freight and the enhancement of the European multimodal TEN-T (passenger and freight) rail corridors [3]. To this end, the European Union is making a significant effort through initiatives and European programs such as Shift2Rail/ERJU [4], where, in its multi-annual program [5], among other things, the digitalization of freight transport is addressed to increase its dynamism and efficiency. Improvements through new systems that allow for better inspection and standardization of border control points, the management of goods and compositions in logistics centers and marshalling yards, as well as integration with various control centers and multimodal and last-mile management platforms, will enable a reduction in journeys, making them more flexible and making rail transport more competitive.

In Shift2Rail this problem started to be addressed, and as a result, the European initiative Intelligent Video Gate (IVG) emerged as a solution. The IVG concept began to be developed within the framework of H2020 Shift2Rail initiative in the FR8Hub project [6] (September 2017 to October 2020) and continued in the FR8Rail III project [7] (September 2019 to June 2023). The IVG concept consists in two different parts: i) The structure of the gate to house all the necessary sensors to inspect the trains, and ii) The software developments to process all the data obtained from the sensors and provide useful information. The scope of this work is only focused on the software side. The application of emerging technologies to the IVG has been already studied in [8] and [9], so the different technologies can be grouped in three main categories: 1) Cameras and image processing, 2) Radio Frequency Identification (RFID) readers and tags, 3) Other type of sensors (speed, axle counter, heat, vibrations, lateral forces in boogies, etc).

The work presented in this paper starts from the point where an IVG system is already installed and capturing images, with a focus only on the detection and recognition of UIC (wagon identifier) and BIC/ILU (container identifier) codes on railway wagons and containers. Therefore, the details of the gate installation and the image acquisition process will not be covered in this publication. Specifically, this work is based on the IVG that is already installed in Gothenburg, which uses line scan cameras to provide a black-and-white image per wagon each time a train crosses the gate. The Gothenburg IVG has previously been used for similar studies, all of them within the context of the European Shift2Rail initiative. In [10], all the framework developed to manage the data sharing and the image analysis is presented. As explained in the public project documentation [6], in FR8Hub only classical computer vision (CV) techniques were applied to detect the codes and the recognition was done using the commercial software CarmenOCR [11]. In FR8Rail III project, the scope of that work was expanded while maintaining the CarmenOCR solution for code identification. This time, two DL-based modules were added for classifying the type of container and for binary classification of whether the container was damaged by graffiti or not. These efforts, initiated in Shift2Rail, are now continuing in ERJU's FA5 under the TRANS4M-R project [12].

There are other related works and commercial solutions that are presented more in detail in Chapter II. However,

This work has been supported by the Spanish PLEC2023-010343 project (INARTRANS 4.0) from MCIN/AEI/10.13039/501100011033.

* Affiliation with Transportation Department, Indra Company, San Fernando de Henares, Madrid, Spain. {adiazd,fparrilla,riglesiasa}@indra.es

† Affiliated with Departamento de Electrónica, Universidad de Alcalá (UAH), Alcalá de Henares (Madrid), Spain. {rafael.barea,luism.bergasa}@uah.es

detailed information on these products is often quite limited, particularly regarding potential costs, results, and limitations. Taking the findings from the FR8Hub and FR8Rail III projects as a baseline for this work, our main contributions are: 1) Review of the state-of-the-art. 2) Creation of a database with real images. 3) Detection and recognition method for wagon and container identification codes based on YOLOv8 that achieves better results than the baseline.

II. Related work

Automation for detecting and recognising codes in railway systems is a topic of great interest that has been studied from different technological perspectives. Optical Character Recognition (OCR) and RFID have been identified as two of the key enabling technologies for this purpose [13].

[14] concluded applying fuzzy Analytic Hierarchy Process (AHP) through an empirical study that RFID is the most suitable technology for Keelung Port. However, although RFID can offer advantages such as greater simplicity in the solution, lower computational load, and therefore shorter processing time, this technology requires that both the track and each wagon and container be equipped with a transmitter or receiver, which is beyond the control of the company responsible for the inspection. This is why, in this work, we have focused the code detection and recognition through OCR with CV cameras in an external gate, which is not only minimally invasive to the track and train infrastructure but also provides information for all trains running on the railway tracks and allows for greater flexibility in expanding the system's functionalities in the future.

In this section, the related work of the different topics involved in this work are review: OCR, OCR applied in railway systems, Datasets and IVG systems.

A. OCR

OCR is the process to convert printed text to digital text and has been studied for many years as shown in [15], which in 1992 reviewed the advancements achieved up to that point. OCR is applied to many sectors, but specifically it has been widely used for Automatic License Plate Recognition (ALPR) in road environments ([16], [17], [18], [19]). There are existing methods as EasyOCR or Tesseract OCR which show promising results. EasyOCR is provided as a Python module that makes use of CRAFT [20] algorithm to detect the code area and then a CRNN model for code recognition [21]. Tesseract OCR is provided both as a Python module and as a C/C++ library. Legacy Tesseract OCR works by recognizing character patterns, but newest versions added Long Short-Term Memory (LSTM) neural nets to improve the implementation. [22] presents a comparative between EasyOCr and Tesseract OCR for ALPR, showing how EasyOCR outperforms Tesseract OCR because of the use of DL for character recognition. The main disadvantage of OCR systems based on classical CV techniques is that they require the images with very controlled conditions and the characters to always follow similar patterns to be recognised. This is not a problem in road environments, as license plates are always located in the same places of the vehicles and contain contrasted characters designed to be recognised by OCR systems but, in railway environments, an OCR system has to deal with codes that are not properly illuminated, located in different parts of wagons and containers, irregular formats and sizes, segmentation uncertainties, damaged characters, and the presence of shadows and occlusions.

B. OCR in Railway Systems

There are not many studies that apply OCR in specific railway environments, but there are that apply OCR for the detection of codes on intermodal loading unit (ILU) containers in loading ports, which are the same containers used on freight trains. There are some of them that apply classical CV techniques for code detection and recognition. [23] uses a CV based method for code detection and Tesseract for code recognition over a dataset of 35 container images, with an overall accuracy of 90.37% and an average processing time of 12.91s per image. [24] applies gray-level feature extraction and gradient based classifier optimization for recognition of container characters, starting from the already isolated area of the code. [25] proposes a CV based container code recognition technique composed of three function modules: namely location, isolation, and character recognition. A text-line region location is used for namely location and trained Support Vector Machines (SVMs) to classify the extracted features for character recognition, obtaining an overall performance over 1214 container images of 91.68% accuracy. [26] is the most promising classical CV based method, as it achieves an overall accuracy of 97.30% with an average processing time of 100ms per image on a computer equipped with 2.2GHz Intel Core i7 over a set of 6000 images obtained from an industrial installation. The method proposes a end-to-end solution that is structured in 4 different steps: i) Region location: The method generates edge images, removes the non-characters and noisy edges and analyse luminance changes to locate character-line regions, ii) Character detection: Multi-scale character detection via sliding window classification is performed based on HOG descriptor and non-maximal suppression (NMS), iii) Character classification: Sobel operator is used to generate features and feed them into two SVM classifiers (one for letters and other for digits), iv) Finally, container code rules are used to apply ensemble rules and make the code recognition more reliable. There are other works that make use of DL based methods. [27] proposes an end-to-end pipeline that uses Region Proposals generated based on Connected Components (CCs) for text detection in conjunction with Spatial Transformer Networks (STNs) for text recognition, but the results provided have been taken from a very small sample of 19 test images. [28] combines both CV and neural networks for detection, and both character segmentation and end-to-end for character recognition. The system is tested over 200 container images obtaining an overall recognition accuracy of 93%. [29] exploits Faster-RCNN [30] to detect and recognise container codes. The method treats each character as a small object to be detected and classified and the results obtained have an overall accuracy of 97.71% over a dataset with 831 container codes. However, the method is tested on a server equipped with a Tesla K80 GPU with 11.92GB of RAM and takes approx. 1.2 seconds per container image, so the method is not applicable to a real-time system. [31] proposes an adaptive deep learning framework for container code localization and recognition. The results are tested over 3000 container images that have taken under different conditions throughout an entire year. The processing seed reaches 1.12 frames/s and the recognition accuracy is 93.33%. Finally, [32] has been the most recent work found regarding the detection and recognition of container codes. The work presents a DL based solution for localizing and recognizing shipping container codes, leveraging ResNet [33] and U-Net [34] for code localization, and a recognition model based on CNN [35] and RNN [36]. The system showcases an overall accuracy of 95% over 256 container images for code detection and 23000 scene text images for text recognition. [37] is the only work that has been found about wagon code detection. It applies Scale Invariant Feature Transform (SIFT) [38] method to find keypoints and identify areas containing wagon codes. The results provided 94.94% accuracy in the best case over a dataset of 79 images, but the study only covers code detection and not code recognition.

The analysis shows that there are many previous works that have attempted to solve the container code detection and recognition problem, but almost no one in railway environments nor wagon codes. Although there are very promising works, the results have been evaluated on small datasets with different characteristics, which makes it very difficult to rigorously compare the results of one work to another. Many of the works have also been evaluated under specific conditions and the information provided about the techniques and the evaluations is not detailed enough to replicate them in the context of a real railway environment. Nevertheless, from the analysis can be concluded as a reference that best systems achieve an accuracy above 90% in container code detection.

C. Datasets

A public dataset that covers all the functionalities of this work, detection and recognition of wagon and container codes in freight train images, has not been found. However, there are some datasets that can serve as a reference for the different parts of the problem separately. [39] and [40] are object detection datasets in which some of the labelled images contain trains. [41] provides 387 container code images in which each character is labelled as a different object. [42] provides 186 wagon images with a wagon code labelled, but the problem with this dataset is that the train models are old, so neither the wagon models nor the wagon codes follow current standards. [43], [44], [45] and [46] are datasets of text detection and recognition in natural images. The analysis shows that almost none of the datasets are railway systems specific, and no one covers all the functionalities required in this work. For this reason, we have created a new dataset based on the images obtained from the Gothenburg's IVG.

D. IVG Systems

There are no public works that cover anything similar to an IVG system but the Shift2Rail previous works ([8], [10]). In contrast, there are many commercial solutions that approach the IVG topic. Nevertheless, information about these products is often quite limited, especially concerning methods, evaluation processes, costs and limitations. Some of the most relevant commercial solutions are: ASE Numbercheck OCR Gate for trains and trucks [47], CAMCO Technologies Rail OCR [48] and the Adaptive Recognition Software application CarmenOCR [11], that is the software used in previous Shift2Rail works and whose results have been used to compare this work.

III. Method

We propose a method to detect and recognise both wagon and container identification codes. The method is based on YOLOv8, a state-of-the-art real time object detection and image segmentation model. This is the last stable version released by Ultralytics of the previous YOLO (You Only Look Once) work [49]. First, a custom YOLOv8 model is used for detecting the codes and cropping the areas. In a second step, specific YOLOv8 models are used for wagon and container codes respectively for detecting each character as an object, and finally, a post processing module is applied to compose the code and assert the code is correct (see Fig. 3).

Wagon identification codes, also known as UIC codes, are defined according to the International Union of Railways (Union internationale des chemins de fer [50]). The UIC code is composed of 12 digits and made up of 5 blocks of numbers separated by spaces or hyphens, as shown in Fig. 1.

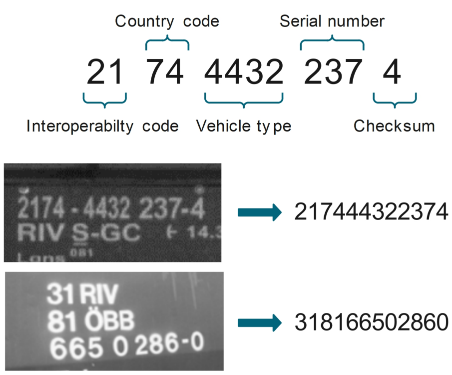

Fig. 1. UIC Code structure and examples

Container identification codes are BIC (Bureau International des Containers) and ILU (Intermodal Loading Unit) codes. BIC code is used to identify standard containers according to ISO 6346. ILU code [51] is compatible with BIC code for ISO 6346 containers, but it can also be used for other European transport units as trailers and swap bodies, according to the EN 13044-1 standard. For the identification purpose of this work, both codes are indistinctly identified as both have the same structure of 11 alphanumeric characters made up of 3 blocks of alphanumeric characters (see Fig. 2), so the container codes will be referred as BIC/ILU in this work.

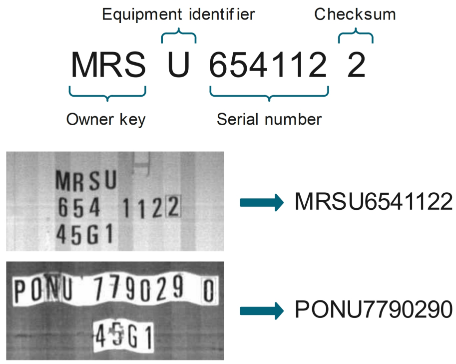

Fig. 2. BIC/ILU Code structure and examples

The method is based on YOLOv8 for both code detection and recognition. YOLOv8 is provided as a Python module and allows for a straightforward approach to model training and inference. In this work, a first model has been trained with freight train images for UIC and BIC/ILU code detection. Then, that identified regions are passed to a second step where there are two different models for UIC and BIC/ILU character recognition that have been trained with images of that code regions. The UIC and BIC/ILU models have been trained separately for each case to detect each alphanumeric character as a different object. Finally, the results of character recognition are passed to a post processing module that composes the code and checks the integrity of the detection using the checksum number Fig. 3.

IV. Experimental Results

As shown in Chapter II, previous works use small and not standardized datasets, and there is no one that covers both wagon and container codes, so for this reason we had to create our own dataset. In this work, we had the privilege to access images from Gothenburg IVG installed in previous Shift2Rail works (private dataset). The obtained images are per wagon, black-and-white and with resolution enough to apply code recognition. There are images with empty wagons, wagons with 1 container and wagons with up to 4 containers. So, in each image there are 1 UIC code and 1, 2, 3 or 4 BIC/ILU codes. In Fig.4, an example image of a wagon with 2 containers is shown. The datasets that have been generated to train and evaluate the models consists of 1000 wagon images for code detection and two different datasets composed by the code region crops of the detected codes for both BIC/ILU and UIC recognition. Only 1000 images were used despite being able to obtain more, as it was detected that after a certain period of time the same trains were passing by again due to their scheduled periodic routes, making most of the images obtained practically identical. This is one of the improvement points identified by having only one IVG, as it would allow for a more generalized and robust system if images could be obtained from different IVGs.

To compare the work proposed with the most significant state-of-the-art system capable of detecting and recognizing wagon and container codes, the previous system from Shift2Rail works based on CarmenOCR software has been also implemented in the experiments as baseline. The models of the method proposed have been trained with 900 images for code detection and the corresponding crop regions for code recognition. Both systems have been evaluated over the same 100 test images, containing a total of 100 wagons and 149 containers. The evaluation was conducted for each code by checking each complete detected code against every complete existing code in the image. In cases where the code is damaged, partially occluded, or not human readable, the code has been labeled as ”Unknown” in the ground truth (GT), so the system should indicate the presence of a code even if it cannot read it. The possible results for each code are: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), considered as shown in Table I.

	Code to detect	Code detected
TP	Code Number or Unknown	Code Number or Unknown
TN	None	None
FP	None	Code Number or Unknown
FN	Code Number or Unknown	None

TABLE I
CONSIDERED RESULTS IN EXPERIMENT

These are the metrics that have been evaluated:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Misclassification = \frac{FP + FN}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Sensitivity(Recall) = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

In the evaluation process, only correct or incorrect codes have been considered, not considering partial code recognition nor the confidence of the detection. The experimental results for both approaches are shown in Table II.

	UIC		ILU	
	Our method	CarmenOCR	Our method	CarmenOCR
Accuracy	**0.74**	0.32	**0.94**	0.42
Misclass.	0.26	0.68	0.06	0.58
Precision	1.0	0.97	1.0	0.76
Sensitivity	0.73	0.29	0.93	0.31
Specificity	1.0	0.8	1.0	0.73
TP	70	28	115	38
TN	4	4	37	32
FP	0	1	0	12
FN	26	68	9	86

TABLE II
COMPARISON OF UIC AND ILU BETWEEN OUR METHOD AND CARMENOCR

We can observe how our approach based on YOLOv8n outperforms in all the metrics the commercial solution CarmenOCR. The overall accuracy obtained for our approach is

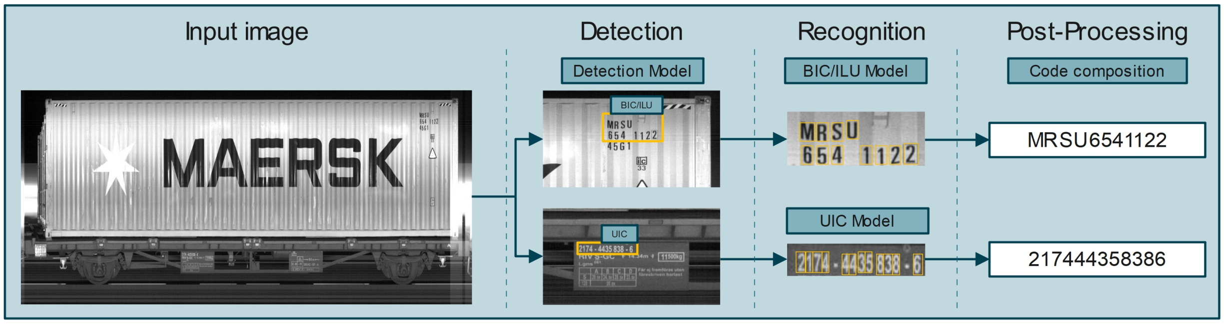

Fig. 3. Method workflow

Fig. 4. Example image of a wagon with 2 containers

74% in UIC codes (32% for CarmenOCR based approach) and 94% in BIC/ILU codes (42% for CarmenOCR based approach). The results in detecting UIC codes are significantly worse compared to BIC/ILU codes. This is due to the greater irregularity in the format and the surface where the codes are located, which makes their recognition more difficult. Fig.5, shows 2 example images of different detections by Our Approach and CarmenOCR Approach for the same UIC codes. In the first image, Our Approach identifies the code but CarmenOCR Approach doesn't even detect it. This may be due to the upper part of the code being partially occluded, which can make detection difficult. In the second image, Our Approach also identifies the code, while CarmenOCR Approach incorrectly identifies another code that is on the wagon. Wagons and containers have multiple codes, as previously mentioned, which makes the detection problem more complex and can lead to this type of erroneous identifications. Precision

Fig. 5. Example of 2 different UIC detections

and specificity are 1.0 in all the cases for the method proposed as there are 0 FP. This is because we use the last digit of the code, the checksum number, to check that the code obtained is correct. In case the checksum number is wrong, the detection is tagged as “Unknown”. Some of these detected “Unknown” are due to the detected code is damaged or occluded, so the code cannot be properly recognised, or just due to the system has not been able to do the recognition step. CarmenOCR does not have any implementation to detect “Unknown” codes, so that causes that CarmenOCR do have some FP, and less TP detections in the cases of “Unknown” that are marked as so in the GT.

V. Conclusions and Future Works

In this work, a method has been proposed to solve the problem of automating code recognition in freight trains using DL-based techniques. An analysis of related previous works has been conducted, concluding that there is a lack of standardization in the methodology for applying and evaluating the proposed methods. This is partly due to the difficulty of accessing real railway systems for testing, as it is not trivial to obtain images of this type to test and evaluate the methods. The system has been compared with the only previous work that met similar functionalities, and better results were obtained in all aspects by evaluating both systems with the same images and under the same conditions. The improvements are **+231% in UIC** codes and **+224% in BIC/ILU** codes. Additionally, in this new proposed method, a final digit-checking step has been implemented, which reduces false positives and provides identification information for codes that could not be recognized.

For future work, we want to expand the dataset with images from different portals, under various weather and lighting conditions, and with images taken from different perspectives, so that the dataset is more generalized and thus allows for better training of more robust DL models. We also want to keep improving the techniques used and expanding the system's functionalities, so that it not only detect UIC and BIC/ILU codes but also cover more identified needs in railway systems.

References

[1] “European Green Deal.” https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_es.

[2] “Fostering Railway Sector through European Green Deal.” https://www.era.europa.eu/system/files/2022-10/fostering_railway_sector_through_european_green_deal_en.pdf.

[3] “European Multimodal TEN-T.” https://transport.ec.europa.eu/transport-themes/infrastructure-and-investment/trans-european-transport-network-ten-t_en.

[4] “Europe's Rail Mission and Objectives.” https://rail-research.europa.eu/about-europes-rail/europes-rail-mission-and-objectives/.

[5] “Europe's Rail Joint Undertaking Multi-Annual Work Programme.” https://rail-research.europa.eu/wp-content/uploads/2022/03/EURAIL_MAWP_final.pdf.

[6] "Shift2Rail FR8Hub Project." https://projects.shift2rail.org/-s2r_ip5_n.aspx?p=FR8HUB.

[7] "Shift2Rail FR8Rail III Project." https://projects.shift2rail.org/-s2r_ip5_n.aspx?p=FR8RAIL

[8] B. Kordnejad, B. Mitrovic, M. Aronsson, J. Bergstrand, and M. Åkerfeldt, "Intelligent video gate–a conceptual application of emerging technologies in rail freight transports," *Transport Research Arena TRA 2020, April 27-30, 2020, Helsinki, Finland*, 2020.

[9] B. Mitrovic, "The effects of emerging technologies in rail yards and intermodal terminals," 2019.

[10] B. Kordnejad, M. Kjellin, M. Aronsson, G. R. Garcia, S. C. Vilabella, R. Wohlrath, I. Nordmark, R. Lengu, M. Åkerfeldt, and J. Bergstrand, "Intelligent video gate–automated detection of wagons and intermodal loading units for image processing and sharing and exploitation of data," in *Proceedings of the 13th World Congress on Railway Research.–2022*, 2022.

[11] "CARMEN OCR – Container Code Recognition." https://adaptiverecognition.com/products/carmen-ocr-container-code-recognition/.

[12] "FP5-TRANS4M-R project plan for 2025." https://projects.rail-research.europa.eu/eurail-fp5/expected-results/.

[13] L. Heilig and S. Voß, "Information systems in seaports: a categorization and overview," *Information Technology and Management*, vol. 18, pp. 179–201, 2017.

[14] S.-L. Chao and Y.-L. Lin, "Applying fuzzy analytic hierarchy process (fuzzy ahp) to select an automatic container number identification system in port terminals," in *2010 8th International Conference on Supply Chain Management and Information*, pp. 1–7, IEEE, 2010.

[15] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of ocr research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

[16] S. Bansal, M. Gupta, and A. K. Tyagi, "A necessary review on optical character recognition (ocr) system for vehicular applications," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 918–922, IEEE, 2020.

[17] M. T. Qadri and M. Asif, "Automatic number plate recognition system for vehicle identification using optical character recognition," in *2009 international conference on education technology and computer*, pp. 335–338, IEEE, 2009.

[18] E. K. Kaur and V. K. Banga, "Number plate recognition using ocr technique," *International Journal of Research in Engineering and Technology*, vol. 2, no. 09, p. 286290, 2013.

[19] C.-N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 1, pp. 59–67, 2014.

[20] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9365–9374, 2019.

[21] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[22] D. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun, "Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pp. 966–971, IEEE, 2022.

[23] C.-S. Fahn, B.-Y. Su, and M.-L. Wu, "Vision-based identification extraction for freight containers," in *2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp. 51–57, IEEE, 2015.

[24] M. Goccia, M. Bruzzo, C. Scagliola, and S. G. Dellepiane, "Recognition of container code characters through gray-level feature extraction and gradient-based classifier optimization.," in *ICDAR*, p. 973, 2003.

[25] W. Wu, Z. Liu, M. Chen, X. Yang, and X. He, "An automated vision system for container-code recognition," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2842–2855, 2012.

[26] Y. Li, H. Li, and G. Gao, "Towards end-to-end container code recognition," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 15901–15918, 2022.

[27] A. Verma, M. Sharma, R. Hebbalaguppe, E. Hassan, and L. Vig, "Automatic container code recognition via spatial transformer networks and connected component region proposals," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 728–733, IEEE, 2016.

[28] Y. Liu, T. Li, L. Jiang, and X. Liang, "Container-code recognition system based on computer vision and deep neural networks," in *AIP Conference Proceedings*, vol. 1955, AIP Publishing, 2018.

[29] W. Zhiming, W. Wuxi, and X. Yuxiang, "Automatic container code recognition via faster-rcnn," in *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 870–874, IEEE, 2019.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[31] R. Zhang, Z. Bahrami, T. Wang, and Z. Liu, "An adaptive deep learning framework for shipping container code localization and recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2020.

[32] S. Hlabisa, R. L. Khuboni, and J.-R. Tapamo, "Automated shipping container code localization and recognition using deep learning," in *2024 Conference on Information Communications Technology and Society (ICTAS)*, pp. 198–203, IEEE, 2024.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[37] U. Sanver, "Identification of train wagon numbers," in *Proceedings of the 2014 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference*, pp. 63–68, IEEE, 2014.

[38] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[40] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[41] "Dataset – Container Code." https://universe.roboflow.com/-ztjdhbuomgsmvpdx-jdvpuoaupjuolfys/ex-mda3v.

[42] "Dataset – Wagon Code." https://universe.roboflow.com/sedykh-marat-dxrw3/wagon-numbers-detection/dataset/2,.

[43] N. Nayef, Y. Patel, M. Busta, P. N. Chowdhury, D. Karatzas, W. Khlif, J. Matas, U. Pal, J.-C. Burie, C.-l. Liu, *et al.*, "Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019," in *2019 International conference on document analysis and recognition (ICDAR)*, pp. 1582–1587, IEEE, 2019.

[44] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," *arXiv preprint arXiv:1601.07140*, 2016.

[45] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 1083–1090, IEEE, 2012.

[46] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

[47] "ASE Numbercheck – OCR Gate for trains and trucks." https://www.ase-gmbh.com/en/services/ocr-gates.

[48] "CAMCO Technologies – Rail OCR." https://www.camco.be/en/solutions/rail-ocr-portal.

[49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[50] "UIC, Home." https://www.camco.be/en/solutions/rail-ocr-portal.

[51] "ILU, Home." https://www.ilu-code.eu/en/.