

Article

Not peer-reviewed version

Automated Container Code Recognition with YOLOv7 and TrOCR

[José António Santos](#)*, [Daniel Duarte Canedo](#), [António José Ribeiro Neves](#)

Posted Date: 11 August 2025

doi: 10.20944/preprints202508.0232.v1

Keywords: Optical Character Recognition (OCR); container code recognition; port operations; YOLOv7; TrOCR



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Automated Container Code Recognition with YOLOv7 and TrOCR

José António Santos *, Daniel Duarte Canedo  and António José Ribeiro Neves 

Institute of Electronics and Informatics Engineering of Aveiro (IEETA), 3810-193 Aveiro, Portugal

* Correspondence: josesantos01@ua.pt

Abstract: Maritime transport plays a pivotal role in global trade, where efficiency and accuracy in port operations are crucial. An aspect of the large scale port operations is container code recognition. Manual inspection of container codes leads to delays and errors, prompting the need for intelligent, automated solutions. This article proposes a hybrid approach to Optical Character Recognition systems that integrates YOLOv7 for text detection with the TrOCR transformer-based model for the recognition of the container codes alphanumeric characters. The systems are designed to address key challenges encountered in real-world port environments, such as varying lighting conditions, image distortions and variable container code orientations. A comprehensive evaluation was conducted utilizing datasets that reflect the diverse conditions of port operations, including different orientations and lighting scenarios. The results indicate that this hybrid approach achieves high precision in both text detection and recognition tasks, outperforming traditional OCR tools like Tesseract and EasyOCR. The integration of YOLOv7 and TrOCR allows for end-to-end processing with improved robustness and accuracy. By handling multi-orientation container codes, the proposed system demonstrates an advancement in automatic code recognition, offering a scalable solution for enhancing operational efficiency in port and freight logistics.

Keywords: Optical Character Recognition (OCR); container code recognition; port operations; YOLOv7; TrOCR

1. Introduction

The use of maritime transport has experienced significant growth in recent decades, becoming the backbone of global trade, accounting for approximately a 80% share of global trade in 2024 [1]. This increase comes from its cost effectiveness compared to air or land transport, as well as the accelerated globalization of supply chains. This growth has had an impact on port infrastructures, which are forced to adapt continuously to ensure smooth operation of supply chains. As critical nodes in global logistic networks, ports face several challenges where a single one can lead to cascading delays, causing increased operational costs in several industries.

Given that port operations involve complex processes that require precise coordination to maintain efficiency, there has been a demand for faster turnaround times and higher throughput. The implementation of industry 4.0 technologies in these scenarios has been a growing trend, creating a digital transformation, known as 'Smart Ports' [2]. These smart ports are characterized by the use of newer technologies, such as the Internet of Things (IoT), Machine Learning (ML), and Big Data [3]. These technologies are used in all phases of port operations [4], ensuring efficient and optimized operations.

The applications of ML usage in port operations are diverse, reflecting the wide range of ML and Deep Learning (DL) algorithms [3]. Within these, common implementations include predictive maintenance of equipment, scheduling systems, automated stacking cranes, and numerous other solutions. Among these, Optical character Recognition (OCR) systems have garnered particular attention for their impact in this environment. By enabling automatic extraction of textual information, OCR proves

indispensable in several systems, many of which are currently used as industry practices [5]. OCR has proven highly effective in yard operation, such as hazard placard detection and container code recognition. These applications improve the efficiency of container handling equipment, leading to shorter turnaround times. Although such technical advancements bring ports closer to an Industry 4.0 standard, there remain an ongoing need to develop more accurate and robust systems to fully meet the operational demands of modern maritime logistics.

To address these challenges of container code recognition, this work proposes a novel OCR system optimized for port operations. Existing OCR solutions like Tesseract [6], initially designed for digitized documents, and EasyOCR, optimized for generalized real-world text, perform well in their respective domains [7] but often fail under specific conditions such as occlusions, reflections, rust, and other factors that degrade accuracy. The proposed system is designed to handle unique challenges, such as varying orientations and conditions, without the need for explicit rotation correction or orientation normalization. Rather than developing an entirely new end-to-end framework, this work adopts a hybrid approach, integrating state-of-the-art detection and recognition models into a streamlined pipeline. This architecture enables the system to maintain high accuracy and robustness in port environments, ensuring reliable performance even under challenging conditions.

2. Materials and Methods

2.1. Literature Review

As the need for parsing the written information into machine-readable formats increased, the demand of OCR systems followed suit. As the use of ML and DL technologies became more prevalent [8], the need for an effective OCR system became apparent. The development of OCR systems has evolved into "end-to-end" tools, where the user is able to input an image and receive machine-readable text as output, without the need for any additional processing steps. Typically, modern OCR pipelines consist of two main stages: text detection, to localize text regions in an image, and text recognition, to convert detected text into machine-readable characters. Despite some models were created for specific tasks, such as the recognition of digitized characters [6], most OCR systems typically lack the ability to handle text in different orientations, such as vertical text, which is common in port operations.

Most of common detection models include Character Region Awareness for Text detection [9] (CRAFT), and You Only Look Once [10] (YOLO). While these models achieve strong performance, modifying the architecture can lead to inference speed or detection accuracy gains [12]. Another common optimization approach is to fine-tune the model on task-specific data, allowing the model to adapt to the specific characteristics of the text in that domain.

The recognition stage typically involves the use of recurrent neural networks (RNNs), Long short term memory networks (LSTMs), and the incorporation of Convolutional Neural Networks (CNNs) with RNNs to create CRNNs. The approach of interconnecting techniques is common, as each technique has its own advantages and disadvantages [11]. As in the case of modified algorithms, depending on the task at hand, their use can lead to a faster inference speed [12].

New advances in the recognition stage utilize transformer-based models, introduced by Vaswani et al. [13], have excelled in various natural language processing and computer vision tasks. These transformers are characterized by their encoder-decode structure, relying on positional encoding and feed-forward layers without convolution or recurrence. With this foundation, following research has introduced several enhancements to the original architecture, such as memory compression and query prototyping in order to reduce attention complexity [14].

These developments have allowed this technology to be used in the field of OCR. The Transformer Based OCR (TrOCR), developed by Li et al. [15] utilizes a Vision transformer (ViT) encoder and a text decoder, which allows it to learn visual patterns without relying on convolutional layers. This architecture has shown itself effective in several scene text benchmarks and while not having complex pre-processing and post-processing steps, it makes it easier to deploy in real-world scenarios.

The literature review highlights the evolution of OCR systems, from traditional methods to modern deep learning approaches. The integration of transformer-based models, such as TrOCR, represents a significant advancement in the field, offering improved performance and flexibility. Our proposal builds upon these advancements to develop a robust OCR system tailored for container code recognition in port operations.

2.2. Dataset

2.2.1. ISO 6346 Standard

In the context of creating a container code recognition system, the model must be able to detect and recognize the code of each container. These codes have to follow a standard structure, which is defined by the International Organization for Standardization (ISO) 6346 standard [16]. This standard was developed by the ISO and the register managed by the Bureau of International Containers (BIC) [17]. In order to be utilized in the global supply chain, all containers must adhere to this standard, ensuring that they are easily identifiable and traceable. This standard ensures a uniformity in global logistic operations, allowing for a more efficient and reliable system. This standard simplifies the process for OCR systems, by providing a clear structure for the codes, which can be used to train the models.

The ISO 6346 defines the structure of the code into two main parts, as seen in Figure 1, the container number and the size and type code. The container number is split into four parts:

1. **Owner code:** Consists of 3 letters.
2. **Category identifier:** Designates the type of container, most common ones are:
 - J: Detachable freight container equipment
 - U: General purpose container
 - Z: Trailers and chassis
3. **Serial number:** Consists of 6 digits, being the unique identifier of the container within that operator's fleet.
4. **Check digit:** A single digit, calculated mathematically in order to verify the integrity of the code.

The size and type code is a four-digit code, that provides more information about the container physical characteristics.

1. **1st character:** Overall length.
2. **2nd character:** Overall height.
3. **3rd–4th characters:** Type and additional features (e.g., General-G0 to G3; Tank-T0 to T9).

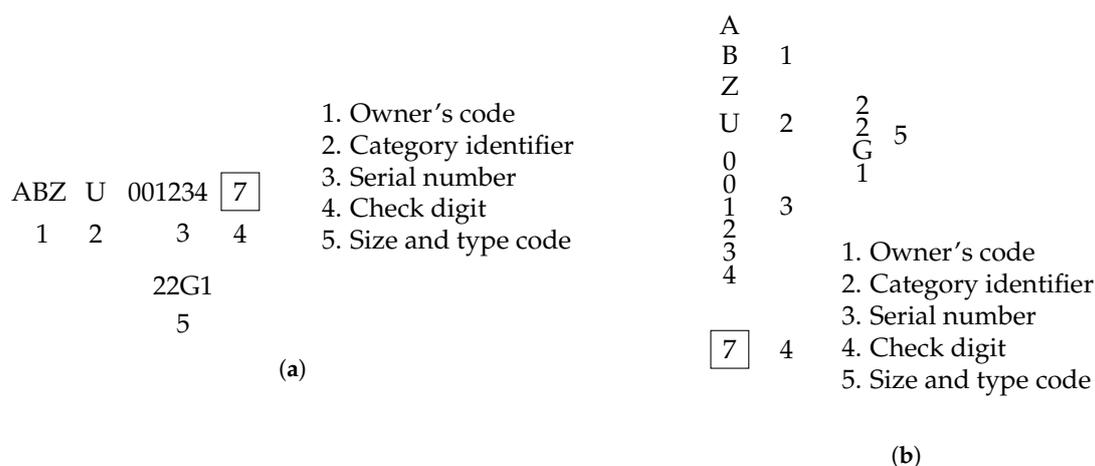


Figure 1. ISO 6346 container code structure. (a) Horizontal layout. (b) Vertical layout.

2.2.2. Dataset Description

To ensure a robust performance of the models across diverse operational environments, this work leverages two distinct datasets. The first dataset is provided by the Port of Sines Authority (PSA),

while the second is open source from GitHub [18]. Each dataset is tailored to specific environments in the container logistics process, providing images at entry and exit points.

The PSA dataset, comprised of 3,295 images, contains captured images from the rail terminal of the Port of Sines. These images were captured while the trains moved in and out of the terminal, where a fixed camera captured the containers during the working hours of the port. This constant monitoring allows the dataset to reflect the realistic challenges in port operations, such as rain, fog and sun glare. This dataset realistically represents port operations, which is why it as served as the primary training and testing set for the model.

As the codes follow the ISO 6346 standard, which allows for both horizontal and vertical orientations, the dataset includes images of both formats. This is illustrated in Figure 2, where 2 (a) shows a horizontal code and in 2 (b) a vertical code.



Figure 2. Example images from the PSA dataset; (a) horizontal code. (b) vertical code.

The open source dataset, available on GitHub, contains 2,910 images depicting truck-based transport environments. These images are focused on two different angles. The first angle is a side profile view of the container, while the second angle collects the rearview of the container, where the container doors are located. While the PSA dataset did not include truck environments, this dataset shows an operational environment that is also present in the Port of Sines infrastructure, making it a ideal supplement to create a model with comprehensive coverage of all port scenarios.

By leveraging a dual dataset approach, the models are trained to perform reliably across a wide range of scenarios, from the controlled environment of the rail terminal to the more variable conditions of truck operations, while ensuring accurate recognition of both vertical and horizontal container codes. The combination of these datasets allows the model to learn and generalize across the dynamic and often unpredictable nature of global transportation systems.

In order to support the training of the models, a unified annotation schema was adopted. Using the VGG Image Annotator (VIA) tool [19], each image in the dataset was manually annotated with bounding boxes around the container codes, using a consistent format, which includes the unique identifier for each container code, the size and type code, obtaining a bounding box and containing text for each region of interest. These annotations are reused in the detection and recognition training, allowing the model to learn from the same data in both stages.

2.3. Hybrid Pipeline Architecture

The proposed OCR system is a structured hybrid pipeline architecture, designed to effectively handle the harsh conditions of port operations. This architecture is divided into two main stages: the detection stage and the recognition stage.

2.3.1. Detection Stage

The detection stage uses the YOLOv7 model, which is known for its high inference speed and accuracy in object detection tasks. While not being a model specifically designed for character detection it has shown strong performance in detecting text in port scene environments, making it suitable for this task. The training of the YOLOv7 model is divided into two distinct phases, each involving separate training processes. These processes are training while the backbone is either frozen or unfrozen.

Given the pre-trained nature of the YOLOv7 model, there is a need to fine-tune it for text detection.

2.3.2. Recognition Stage

The recognition stage employs the TrOCR model, a transformer based model, which has shown strong performance in various OCR tasks, particularly in handling complex scenarios. This robust performance was a key factor in its selection, as the model achieves competitive accuracies on OCR benchmarks [15]. This model utilizes a Seq2Seq training strategy, which is implemented using the Hugging Face Transformers library [20].

2.3.3. Training Protocol

To ensure effective training and evaluation, the datasets were split into training, validation, and testing sets using a 70/20/10 ratio, where 70% of the dataset was used for training, 20% for validation, and 10% for testing. This unified split allows both the detection and recognition models to learn from the same distribution of data.

This approach also ensures that the test set consists entirely of unseen images for both models, allowing for a fair evaluation of their performance.

Hardware and Frameworks

- **GPU:** NVIDIA GeForce RTX 1080 Ti (11GB VRAM)
- **TrOCR Framework:** Hugging Face Transformers v4.46.3
 - Base Model: trocr-base-stage1 from Hugging Face
 - WandB Integration: Full training metrics logging
- **YOLOv7 Framework:** Custom PyTorch implementation
 - Input Resolution: 512x512 pixels
 - Augmentations: Mosaic (1.0), MixUp (0.15), FlipLR (0.5)
 - WandB Integration: Full training metrics logging

Hyperparameter Configuration

The hyperparameters were carefully selected to optimize the training of both models, ensuring effective learning and convergence. The Table 1 summarizes the key hyperparameters used for training both the TrOCR and YOLOv7 models.

Table 1. Comparative training hyperparameters.

Parameter	TrOCR	YOLOv7 (Frozen)	YOLOv7 (Unfrozen)
Batch Size	8	16	16
Learning Rate	5e-5	0.01	0.01
LR Schedule	Linear Warmup	Cosine Annealing	Cosine Annealing
Warmup Steps	500 steps	3 epochs	3 epochs
Epochs	25	50	50
Optimizer	AdamW	SGD	SGD
Momentum	-	0.937	0.937
Weight Decay	0	0.0005	0.0005

TrOCR Specifications

The TrOCR model was trained using a Vision Encoder-Decoder architecture, with the following technical specifications:

- **Architecture:**
 - Vision Encoder-Decoder with 384M parameters
 - ViT encoder processes 384×384 images-16px patches
 - Transformer decoder: 1024 dim, 16 attention heads
 - Maximum sequence length: 64 tokens
 - Gradient clipping at 1.0 norm
 - Max Grad Norm: 1.0
- **Training Configuration:**
 - Mixed FP16 precision (O1)
 - Gradient checkpointing enabled
 - Early stopping on validation CER

YOLOv7 Specifications

As for YOLOv7, the pipeline followed two training variants, one with the backbone frozen and the other with the entire network unfrozen. The characteristics of these configurations follow these technical specifications:

- **Architecture:**
 - Darknet-based with 36.5M parameters
 - Loss components:
 - * Bounding box: 0.05 (CIoU)
 - * Classification: 0.3
 - * Objectness: 0.7
 - Optimal Transport Assignment (OTA) enabled
- **Training Configuration:**
 - FP32 precision
- **Training Variants:**
 - Frozen: First 50 layers fixed
 - Unfrozen: Entire network trainable

The training pipeline used Weights and Biases (WandB) to monitor in real-time the performance of models during training. For the TrOCR recognition model, we tracked the character error rate (CER) and training loss at each step, while for the YOLOv7 detection model, we followed the mean average precision (mAP) metrics at different Intersection over Union (IoU) thresholds. The system automatically saved checkpoints for TrOCR every 500 steps and after each epoch for YOLOv7, allowing us to track progress and recover if needed.

3. Results and Discussion

3.1. Evaluation Metrics

The evaluation of the hybrid OCR system is conducted through a series of metrics that assess both the detection and recognition stages. Due to the different nature of the tasks, distinct metrics are used for each stage. The evaluation metrics for both stages are presented, where the equations for the metrics are also shown.

In the detection stage, precision is used to measure the correctly identified text regions out of all detected regions, while recall measures the percentage of true text regions that were successfully detected. The mean Average Precision (mAP) at Intersection over Union (IoU) thresholds is also used, which provides a comprehensive measure of the model's ability to localize text regions. In this work the $mAP@0.5$ and $mAP@0.5:0.95$ are used, which measure the model's ability to correctly localize text regions at different IoU thresholds.

For the recognition stage, we evaluate character and code level accuracy of the system utilizing Character Error Rate (CER) and Word Error Rate (WER). CER measures the accuracy of individual characters, while WER evaluates the accuracy of the predicted codes as a whole. These metrics are both error based metrics, where a value closer to 0 represents a perfect match between the predicted and ground truth text, indicating no recognition errors occurred.

3.2. YOLOv7 Training

The training of this detection model, two distinct training processes were conducted, utilizing the YOLOv7 architecture. The first phase involved training the model with an unfrozen backbone, allowing the model to adapt its feature extraction capabilities to the container code dataset.

In Figure 3 (a) the unfrozen model training exhibits unstable performance, despite efforts to optimize the hyperparameters. While the precision and recall initially showed signs of learning, after epoch 15 the values began to spike and continued to oscillate significantly, preventing convergence.

By contrast, the frozen backbone training, shown in Figure 3 (b), exhibited a more stable and consistent learning pattern. As the Precision and recall followed a steady upward trend, this suggests that freezing the backbone layers allowed the model to leverage the pretrained feature extraction capabilities, leading to improved performance.

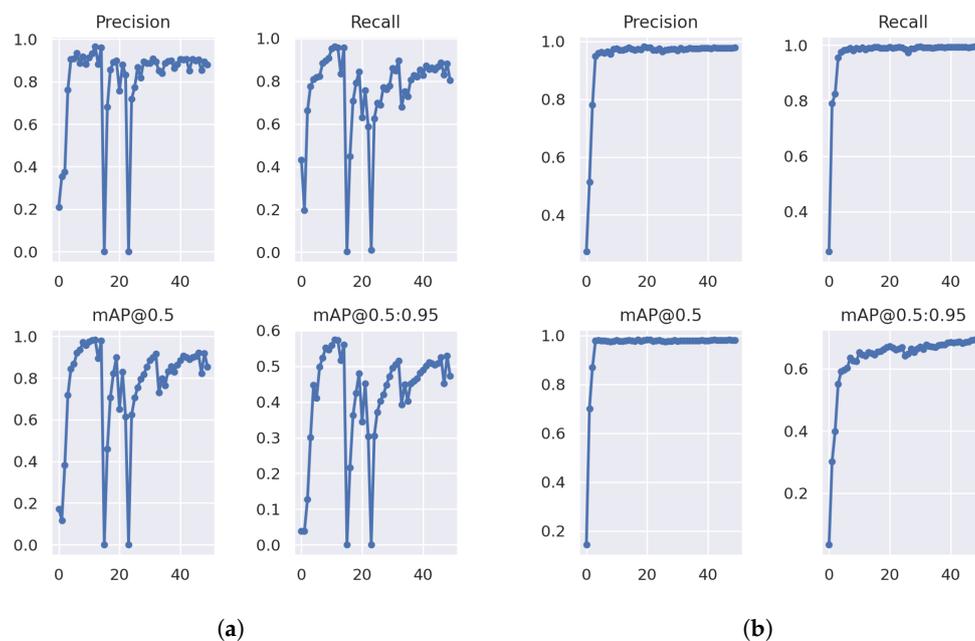


Figure 3. : (a) Yolov7 unfrozen training results. (b) Yolov7 frozen training results.

As shown in Table 2, the frozen backbone model outperformed the unfrozen model across all metrics. The unfrozen models poor performance may stem from insufficient training data, preventing an effective adaptation of its backbone layers. In contrast, the frozen backbone was able to leverage transfer learning more effectively, preserving its robust feature extraction capabilities without overfitting.

Table 2. Comparison of YOLOv7 detection performance with frozen backbone vs. original model.

Model	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
YOLOv7 (Frozen Backbone)	94.8	95.4	96.9	65.3
YOLOv7 (Original)	91.1	79.7	84.0	56.3

3.3. TrOCR Training

The TrOCR model, was fine-tuned using pre-trained weights from the Hugging Face Transformers library, adapting it to recognize both vertical and horizontal container codes. The training leveraged the same dataset employed in the detection stage, ensuring consistency in data distribution and annotation quality.

As shown in Figure 4, the evaluation loss, Eval Loss, and Character error rate (CER) decreased over training steps, indicating effective learning. Given that the CER calculates the amount of correct characters predicted by the model, it was the metric used to evaluate the models' performance. While having some spikes at the 3000 and 4000 steps, the CER metric reached a stable plateau at the end of the training, at 6000 steps.

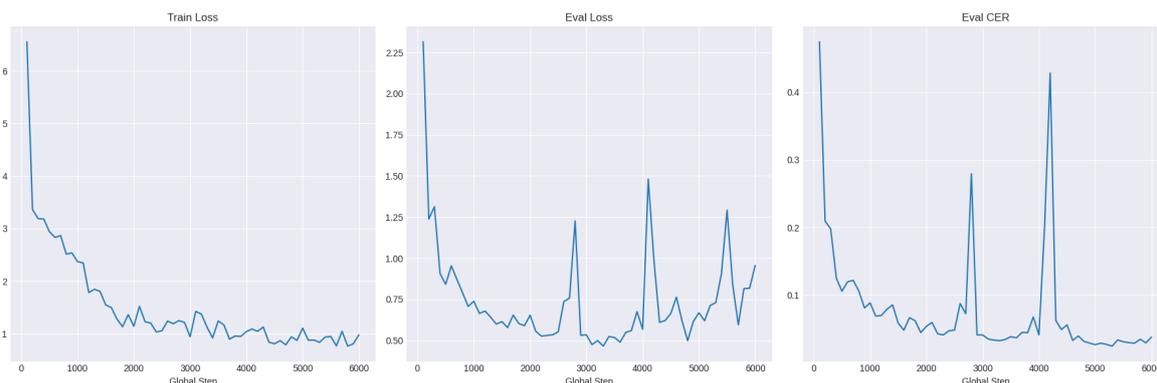


Figure 4. TrOCR training results.

When evaluating on the unseen test set, the TrOCR model demonstrated a strong recognition performance, achieving a CER of 0.0244 and WER of 0.1066. These metrics indicate that the model accurately transcribed fewer than 3% of the characters incorrectly, and successfully recognized entire container codes with An 89.34% exact match rate, as shown in Table 3. These results underscore the models effectiveness in handling the challenges of OCR in container logistics where high accuracy is essential in operational reliability.

Table 3. TrOCR model performance on the test dataset.

Model	CER	WER	Character Accuracy	Exact Match Score
TrOCR	0.0244	0.1066	97.56%	89.34%

3.4. Comparative Results

The final evaluation results of the OCR system, integrating both the detection and recognition stages, are presented in Table 4. For the recognition component, two additional metrics were introduced, the vertical and horizontal recognition accuracy, which respectively measure the code recognition accuracy for vertically and horizontally oriented codes. These metrics are essential, as real work environments may represent either orientation, and evaluating the models' performance in both scenarios is crucial for a comprehensive assessment of the OCR system's capabilities.

Table 4. Final evaluation results of the OCR system using YOLO for detection and TrOCR for recognition.

	Detection Recall	Detection Precision	mAP@50	mAP@95	Vertical Recognition Accuracy	Horizontal Recognition Accuracy
Yolov7+TrOCR	96.77%	99.40%	96.69%	70.84%	99.11%	96.86%
Yolov7+Tesseract	87.66%	99.34%	87.57%	64.41%	2.58%	73.25%
Yolov7+EasyOCR	89.48%	99.54%	89.48%	65.84%	15.31%	64.23%
Tesseract	11.76%	0.41%	2.15%	0.77%	2.04%	33.48%
EasyOCR	88.73%	16.47%	27.09%	10.02%	13.85%	71.17%

The experimental results demonstrate that the combined use of YOLOv7 and TrOCR establishes a robust OCR system for container code recognition. This hybrid system archives an outstanding text detection rate of 96.77% and a high precision of 99.40%, while also achieving a high vertical and horizontal recognition accuracy of 99.11% and 96.86%, respectively.

The ability to accurately recognize both vertical with high accuracy is particularly noteworthy, as conventional OCR systems are optimized for horizontal text due to their training on predominantly horizontal text datasets. As this model successfully generalizes on both text orientations, it eliminates the need for separate models and reducing overall system complexity. This represents a significant advancement for practical applications where text orientation can vary, such as in port operations, barcode scanning and other logistics tasks.

The standalone OCR system reveal limitations that justified the hybrid approach. The tesseract had a poor performance achieving only 0.41% precision in text detection. EasyOCR exhibited inconsistent recognition capabilities, with a 57.32 percentage point gap between vertical and horizontal recognition accuracy, indicating limitations in handling text orientation variability.

As shown in Figure 5, the YOLOv7 and TrOCR hybrid model achieves superior performance across both vertical (a) and horizontal (b) text recognition tasks. This orientation-agnostic performance suggests that TrOCR transformer-based architecture successfully generalizes to rotation variations, a critical capability that is often lacking in traditional OCR systems.



Figure 5. Examples of the hybrid model on the test set. (a) Vertical code recognition. (b) Horizontal code recognition.

4. Conclusions

This paper presents a novel hybrid OCR system that combines YOLOv7 for text detection and TrOCR for text recognition. The integration of these two models address significant challenges in port operations, particularly in detecting and recognizing codes under varying conditions and orientations.

Experimental results demonstrate that the proposed system performs well, achieving high accuracy and robustness compared to existing models. It successfully recognizes text in scenarios where traditional OCR systems struggle, validating its reliability and effectiveness in real-world applications.

The implementation of this hybrid OCR system represents a significant step towards automating and optimizing port operations. By improving the reliability and speed of container code recognition, it contributes to the streamlining of workflows and enabling better resource allocation, leading to improved throughput and operational effectiveness in port settings.

Further improvements to the model's performance can be achieved by exploring different datasets and conditions, as well as investigating the integration of these systems in real-world scenarios. Additionally, real-world deployment should be conducted to evaluate the systems' performance in live operational contexts with comprehensive evaluation metrics to guide interactive refinement. Another promising direction for future research is to extend the systems capabilities and the venerability of the model across different, related environments, such as dry ports being logistics centers outside the port area.

In conclusion, the proposed hybrid OCR system offers an effective solution to the challenge of container code recognition in port operations. With continued development, it has the potential to become a critical component in the modernization and automation of global logistics infrastructure.

Author Contributions: Conceptualization J.S, D.C, A.N; methodology J.S, D.C; software J.S; validation J.S, D.C, A.N; formal analysis J.S; investigation J.S; resources A.N; data curation J.S; writing—original draft preparation J.S; writing—review and editing D.C, A.N; visualization J.S; supervision D.C, A.N; project administration A.N; funding acquisition A.N.

Funding: This study was funded by the PRR—Recovery and Resilience Plan and by the NextGenerationEU funds at Universidade de Aveiro, through the scope of the Agenda for Business Innovation “NEXUS: Pacto de Inovação—Transição Verde e Digital para Transportes, Logística e Mobilidade” (Project no. 53 with the application C645112083-00000059).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The open source data is available at the links referenced in the text. All other datasets referenced in this study are available from the corresponding author. The data is not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AdamW	Adaptive Moment Estimation with Weight Decay
CER	Character Error Rate
CIoU	Complete Intersection over Union
EMA	Exponential Moving Average
FP16	16-bit Floating Point Precision
IoU	Intersection over Union
mAP	mean Average Precision
OCR	Optical Character Recognition
OTA	Optimal Transport Assignment
SGD	Stochastic Gradient Descent

Seq2Seq	Sequence-to-Sequence
TrOCR	Transformer-based Optical Character Recognition
ViT	Vision Transformer
WandB	Weights and Biases
WER	Word Error Rate
YOLOv7	You Only Look Once version 7

References

1. United Nations Conference on Trade and Development. *Review of maritime transport 2024: Navigating maritime chokepoints* (UNCTAD/RMT/2024 and Corr.1). United Nations, 2024. Available online: <https://unctad.org/publication/review-maritime-transport-2024>
2. de la Peña Zarzuelo, I.; Freire Soeane, M.J.; López Bermúdez, B. Industry 4.0 in the port and maritime industry: A literature review. *Journal of Industrial Information Integration* 2020, 20, 100173. <https://doi.org/10.1016/j.jii.2020.100173>.
3. Yang, Y.; Gai, T.; Cao, M.; Zhang, Z.; Zhang, H.; Wu, J. Application of Group Decision Making in Shipping Industry 4.0: Bibliometric Analysis, Trends, and Future Directions. *Systems* 2023, 11, 69. <https://www.mdpi.com/2079-8954/11/2/69/htm>.
4. Filom, S.; Amiri, A.M.; Razavi, S. Applications of machine learning methods in port operations—A systematic literature review. *Transportation Research Part E: Logistics and Transportation Review* 2022, 161, 102722. <https://doi.org/10.1016/j.tre.2022.102722>.
5. Port Equipment Manufacturers Association. Information paper: OCR in ports and terminals (PEMA-IP4). Available online: <https://www.pema.org/wp-content/uploads/2022/09/PEMA-IP4-OCR-in-Ports-and-Terminals.pdf> (accessed on 2025).
6. Smith, R. An Overview of the Tesseract OCR Engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 23–26 September 2007; Volume 2, pp. 629-633. <https://doi.org/10.1109/ICDAR.2007.4376991>.
7. Vedhaviyash, D.R.; Sudhan, R.; Saranya, G.; Safa, M.; Arun, D. Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm. *6th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2022 - Proceedings* 2022, pp. 966-971. <https://doi.org/10.1109/ICECA55336.2022.10009215>.
8. Raj, R.; Kos, A. A Comprehensive Study of Optical Character Recognition. In Proceedings of the 2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES), 2022, pp. 151-154. <https://doi.org/10.23919/MIXDES55591.2022.9837974>.
9. Baek, Y.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character Region Awareness for Text Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019, 2019-June, 9357–9366. <https://arxiv.org/abs/1904.01941v1>.
10. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464-7475. <https://arxiv.org/abs/2207.02696v1>.
11. Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S.J.; Lee, H. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. *Proceedings of the IEEE International Conference on Computer Vision* 2019, 2019-October, 4714-4722. <https://arxiv.org/abs/1904.01906v4>.
12. Feng, X.; Wang, Z.; Liu, T. Port container number recognition system based on improved YOLO and CRNN Algorithm. *Proceedings - International Conference on Artificial Intelligence and Electromechanical Automation, AIEA 2020* 2020, 72-77. <https://doi.org/10.1109/AIEA51086.2020.00022>.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems* 2017, 2017-December, 5999–6009. <https://arxiv.org/abs/1706.03762v7>.
14. Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A Survey of Transformers. *AI Open* 2022, 3, 111-132. <https://doi.org/10.1016/J.AIOPEN.2022.10.001>.
15. Li, M.; Lv, T.; Chen, J.; Cui, L.; Lu, Y.; Florencio, D.; Zhang, C.; Li, Z.; Wei, F. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023* 2021, 37, 13094–13102. <https://arxiv.org/abs/2109.10282v5>.
16. ISO. ISO 6346:2022 - Freight containers — Coding, identification and marking. Available online: <https://www.iso.org/standard/83558.html> (accessed on 2025).

17. Bureau International des Containers. BIC Code - The Standard for Container Identification. Available online: <https://www.bic-code.org/> (accessed on may 2025).
18. Lin, B. ContainerNumber-OCR: Container Number Recognition Based on YOLOv7 and CRNN. GitHub repository, 2023. Available online: <https://github.com/lbf4616/ContainerNumber-OCR> (accessed on February 2025).
19. Dutta, A.; Zisserman, A. VGG Image Annotator (VIA). Version 2.0.12. Available online: <http://www.robots.ox.ac.uk/~vgg/software/via/> (accessed on 30 May 2025).
20. Hugging Face. TrOCR Model Documentation. 2024. Available online: https://huggingface.co/docs/transformers/model_doc/trocr (accessed on March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.