TRUDI and TITUS: A Multi-Perspective Dataset and A Three-Stage Recognition System for Transportation Unit Identification

Emre Gülsoylu[1]
emre.guelsoylu@uni-hamburg.de
André Kelm[1]
andre.kelm@uni-hamburg.de
Lennart Bengtson[2]
lennart.bengtson@uni-hamburg.de
Matthias Hirsch[1]
matthias.hirsch@uni-hamburg.de
Christian Wilms[1]
christian.wilms@uni-hamburg.de
Tim Rolff[3]
tim.rolff@uni-hamburg.de
Janick Edinger[2]
janick.edinger@uni-hamburg.de
Simone Frintrop[1]
simone.frintrop@uni-hamburg.de

[1] Computer Vision Group
University of Hamburg
Hamburg, Germany

[2] Distributed Operating Systems Group
University of Hamburg
Hamburg, Germany

[3] Human-Computer Interaction Group
University of Hamburg
Hamburg, Germany

Abstract

Identifying transportation units (TUs) is essential for improving the efficiency of port logistics. However, progress in this field has been hindered by the lack of publicly available benchmark datasets that capture the diversity and dynamics of real-world port environments. To address this gap, we present the TRUDI dataset—a comprehensive collection comprising 35,034 annotated instances across five categories: container, tank container, trailer, ID text, and logo. The images were captured at operational ports using both ground-based and aerial cameras, under a wide variety of lighting and weather conditions. For the identification of TUs—which involves reading the 11-digit alphanumeric ID typically painted on each unit—we introduce TITUS, a dedicated pipeline that operates in three stages: (1) segmenting the TU instances, (2) detecting the location of the ID text, and (3) recognising and validating the extracted ID. Unlike alternative systems, which often require similar scenes, specific camera angles or gate setups, our evaluation demonstrates that TITUS reliably identifies TUs from a range of camera perspectives and in varying lighting and weather conditions. By making the TRUDI dataset publicly available, we provide a robust benchmark that enables the development and comparison of new approaches. This contribution supports digital transformation efforts in multipurpose ports and helps to increase the efficiency of entire logistics chains.

1 Introduction

Multipurpose terminals in ports need to accommodate various modes of transportation and volatile cargo volumes with limited available space. The resulting dynamic environments created by constant modifications in storage configurations and operational processes make the use of fixed infrastructure for monitoring terminals impractical. This leads to increased manual labour for inventory keeping and inadequate traceability of transportation units (TUs), such as containers and cranable semi-trailers. Efficient, digital terminal monitoring thus requires the reliable identification of individual TUs [28, 33].

TUs have standardised dimensions and feature unique markings as defined in ISO6346 [4]. These alphanumeric ID codes ensure unambiguous visual identification and are usually painted on the top or sides of each TU. They consists of a four-letter owner code, a six-digit serial number, and a single check digit. Despite advancements in automation systems for identification technologies like Optical Character Recognition (OCR) gates [43] and RFID tags [34] on the TUs, these approaches often fall short in adaptability, especially in multipurpose terminals with seasonal operational variability. Existing solutions for TU identification typically rely on character detection from a specific target area followed by character recognition [9, 43]. Other solutions relax the assumption of a predefined target area by adding another step to detect the ID first [38, 40]. However, these methods are still constrained to fixed camera placements and struggle with changing perspectives, particularly when using mobile cameras mounted on unmanned aerial vehicles (UAVs) or reach stackers (RSs) [36]. Moreover, existing methods are mostly evaluated on proprietary datasets often consisting of images taken in one single port which limits comparability. There are currently no publicly available datasets for TU identification [23].

We introduce the TRansportation Unit Detection and Identification (TRUDI) dataset comprising images captured from both aerial and ground perspectives under different lighting and weather conditions. This dataset addresses the comparability issue and supports the progress towards more adaptable port monitoring operations utilising moving cameras. Moreover, we propose a flexible TU identification method which is suitable for use with both aerial and ground-based imagery and, thus, does not rely on fixed infrastructure. We introduce and employ the Three-stage Identification of Transportation UnitS (TITUS) pipeline that consists of segmentation of TU instances, ISO6346 compliant ID text detection, and text recognition. The use of an instance segmentation stage prior to text detection and text recognition enables associating TUs with their IDs reducing the search space for text detection. Additionally, the association of the segmented instances and ID codes can support the localisation of TUs inside terminals using mobile cameras with GPS sensors. This facilitates down-stream applications such as the creation of a digital twin for the detailed analysis of operational processes. With the release of TRUDI and the introduction of TITUS, we aim to provide researchers and practitioners with valuable resources for developing and evaluating new methods on multiperspective and robust identification of TU.

In summary, our contributions are: (1) a new and publicly available dataset[1], TRUDI, for TU identification from aerial and ground perspectives, (2) a novel three-stage pipeline, TITUS, and the (3) detailed evaluation of the proposed pipeline on the TRUDI dataset.

[1] https://github.com/egulsoylu/trudi

2 Related Work

Existing literature mostly focuses on automatic container code recognition using fixed cameras [1, 23, 39], thereby excluding the use of mobile cameras, including ground-based handheld devices and aerial platforms such as UAVs. While the use of UAVs, in particular, enables more scalable and flexible image acquisition, it also introduces new challenges, such as the detection of small text areas within complex backgrounds. These challenges can reduce performance in identifying TUs [36]. Furthermore, the identification of intermodal loading units, such as cranable semi-trailers, remains largely overlooked in the literature [42].

Early approaches for this task often rely on digital image processing techniques [14, 26, 31]. These conventional methods are still employed alongside deep learning (DL) methods to form hybrid solutions, allowing for effective task-specific feature engineering. Nguyen et al. [29] employ both conventional computer vision and machine learning (ML) techniques including histogram of oriented gradients and support vector machines in a pre-processing-intensive method for text detection and recognition. While their approach has a robust pre-processing stage, the lack of comparative evaluation with DL models presents limitations, particularly regarding the adaptability and accuracy of their system in uncontrolled environments. Additionally, Hsu et al. [10] employ YOLOv4 for the initial detection phase and use Tesseract OCR [35] for text recognition, integrating histogram equalisation and morphological operations in the pre-processing stage. Although this method benefits from powerful OCR capabilities, its performance and robustness in varying environmental conditions remain unclear due to limited dataset diversity. Another hybrid method [15], an end-to-end recognition system, applies edge detection and component analysis to classify characters with support vector machines. This approach demonstrates an alternative to DL-based methodologies yet remains susceptible to variations in lighting and TU condition.

Since ML-based solutions have demonstrated automatic feature learning and extraction from a given dataset, researchers have increasingly focused on these methods in recent years. An approach by Zhao et al. [43] introduces the Practical Unified Network (PUN), designed to localise and recognise arbitrary-oriented container codes, integrating detection and classification within a single framework. This model uses a ResNet18 [7] backbone and demonstrates superior performance over traditional CNN- and transformer-based methods such as EAST [44], DETR [45] and ABCv1 [24]. Its end-to-end design provides an efficient solution for static camera settings. However, it is less efficient for mobile cameras which can operate in a perception-action loop [5] to iteratively select better perspectives for image capturing and, thus, reduce the number of unsuccessful text ID recognition attempts. Yang et al. [38] focus on real-time processing with a lightweight model based on multi-reuse feature fusion and a multi-branch structure merger. For this, they optimise detection with MobileOne blocks [37] and recognition using MobileNetV3-small [13]. Even though they demonstrate accuracy improvements over YOLOv5 [6] their methodology does not fully address the challenges involved in the mobile camera-based applications as it treats text recognition as character detection. Character-level detection, however, is not suitable for scene-text detection as the text is scattered in the scene image, and there is no prior information about their location [18]. While the system is capable of high processing speeds, its applicability in real-world conditions may be constrained as the evaluation relies on a non-diverse dataset. Li et al. [23] tackle TU identification as a character detection problem and introduce ACCR-YOLOv7 incorporating a feature extraction module called G-ELAN and an improved Efficient Spatial Pyramid Pooling Module. This model reduces computational complexity by replacing YOLOv7's ordinary 3x3 convolution with GSconv in the neck.

In summary, while significant progress has been made for the transportation units (TUs) identification task, developing robust and adaptable solutions with consistent performance across diverse operational scenarios, environments and perspectives remains a challenge. Current systems are unusable with vehicle-mounted cameras as they require TUs to be placed in a predefined area. Although there are a few publicly available datasets such as Ship Container Code [12], or Container Number-OCR [20] none of them adopted as a benchmark dataset due to low diversity they offer. The lack of benchmark datasets hinders the comparability of the proposed methods. This leads to very high accuracy values reported for methods using less complex datasets [23] and low numbers for methods evaluated on complex datasets [36]. Therefore, TRUDI can serve as a benchmark to enhance the comparability of proposed solutions and facilitate further innovation in TU identification.

3 TRUDI Dataset

To address the lack of publicly available datasets suitable for TU identification using images from mobile cameras, we collected and annotated a comprehensive and multifaceted dataset featuring images captured from both aerial and ground-based perspectives. As shown in Table 1, the dataset comprises 35,034 labelled instances of TUs and their markings, with an average of approximately 48 instances per image. 17,604 instances were collected from ground perspective through various devices, including smartphones, digital single-lens reflex cameras (DSLR), and camera-equipped vehicles such as terminal trucks and RSs. These images were captured during the vehicle's active use in port operations. The remaining 17,430 instances were captured by UAVs using models like the DJI Mavic Pro 3, DJI Mini 2, and DJI Air 3[2]. Sample images from the TRUDI dataset are shown in Figure 1 from both ground and aerial perspective.

The images in TRUDI cover a wide range of perspectives, zoom levels, resolutions, and image qualities, which provides a diverse dataset for object detection, instance segmentation, logo detection, text detection, text recognition and text spotting. This diversity enables trained models to handle real-world scenarios by improving their robustness and generalisation capabilities. The objects are labelled as masks belonging to one of the five classes during the annotation process. Three of these classes represent common TU types (*container*, *trailer*, *tank_container*), while the TU markings are represented by two classes (*id_text*, *logo*). Not all TU masks have associated markings due to occlusions or large distances that make IDs undetectable and illegible.

The number of instances is shown in Table 2. All annotators and reviewers who check the quality of the annotations have been provided with comprehensive guidelines to avoid any inconsistencies in annotations that negatively affect the reliability of models [21]. These guidelines outline the annotation process and provide clear definitions for classes of interest and edge cases such as occlusions or damaged and illegible markings[3].

To ensure environmental and temporal diversity, the images in TRUDI were collected over an 18-month period across different countries, at various times of the day, including daytimes, dusk, and night, as well as during different seasons. Furthermore, the images were taken under various weather conditions, ranging from sunny and partly cloudy to overcast, rainy, and snowy. UAVs were operated at various altitudes ranging from 3 to 120 m, with an average flight altitude of approximately 30 m, to represent different scenarios suitable for

[2] dji.com

[3] Further details about the dataset can be found in the supplementary material.

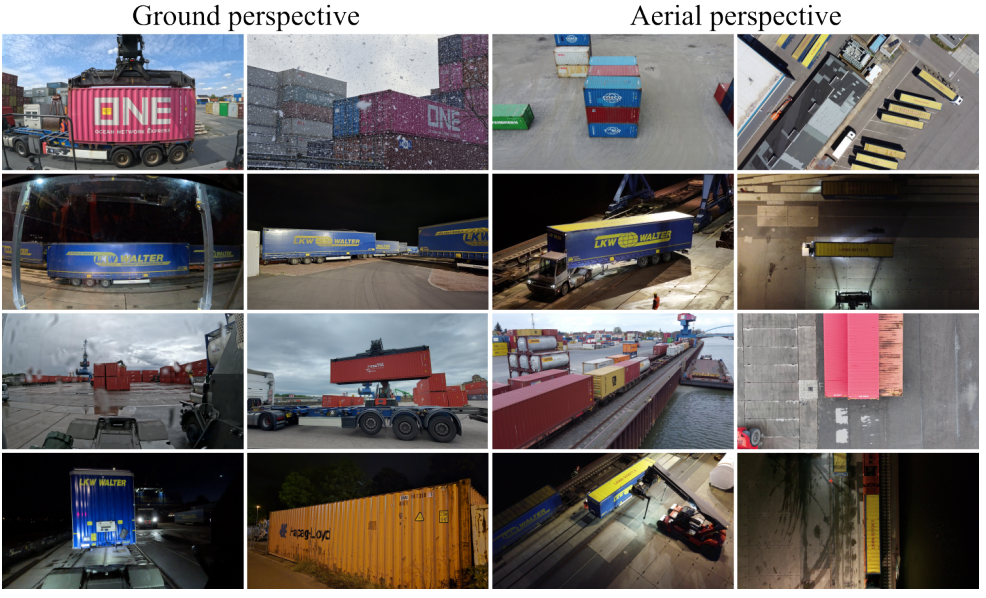

Figure 1: Sample images from the TRUDI dataset showing ground and aerial perspectives.

Table 1: Dataset statistics per perspective and subset. The combined set contains the images from both aerial and ground perspectives.

Perspective	Subset	# Images	# Instances	Avg. Instances	Median Instances
Aerial	Training	231	9864	42.70	24.0
	Validation	77	3869	50.25	24.0
	Test	75	3697	49.29	23.0
Ground	Training	210	9942	47.34	15.0
	Validation	70	3663	52.33	17.0
	Test	70	3999	57.13	21.5
Combined	Training	441	19806	44.91	20.0
	Validation	147	7532	51.24	23.0
	Test	145	7696	53.08	22.0
Total		733	35034	47.80	21.0

Table 2: Class-wise instance count and size categories following COCO style [19] (px).

Category	# Instances	Small ($area < 32^2$)	Medium	Large ($96^2 < area$)
Container	11109	913	4868	5328
Tank Container	808	65	351	392
Trailer	2780	67	455	2258
ID Text	14009	9245	3433	1331
Logo	6328	2096	2361	1871

Figure 2: Overview of the TITUS pipeline. TUs instances are segmented, then the ISO6436 compliant ID texts are detected, finally, a text recognition model extracts detected IDs.

TUs identification missions. We excluded any frames showing the same objects from the identical viewpoint to avoid data redundancy.

Randomly dividing a dataset could lead to a trivial test set or result in certain types of images being included only in one of the subsets. Therefore, we divided our TRUDI dataset into three subsets (60 % train, 20 % validation, 20 % test) while ensuring that the subsets have a similar distribution. The dataset was divided by binning the images based on their brightness, sharpness, and contrast, using a uniform bin range. We chose these features, as they are frequently used in image enhancement to assess image quality [30]. We then combined these bins into a single category for each image. This categorisation enabled us to stratify the subsets effectively.

4 TITUS Pipeline

Our novel system involves three stages: (1) segmenting TU instances (containers, tank containers, and trailers), (2) detecting their ID text area, (3) extracting the ID code from detected text areas and associating the extracted ID with the corresponding TU instance. Figure 2 illustrates this pipeline.

Segmenting TUs instances as a prior step benefits the text detection since it reduces the search space to cropped regions. This increases inference speed and enhances text detection quality by minimising the number of distractors, compared to detecting ID text in the entire image. The resulting cutouts effectively isolate the TUs, avoiding the inclusion of background areas that can still be present in bounding boxes. As a result, IDs from TUs in the background are less likely to interfere with associating each TU to its own ID code, which is especially important in densely packed storage scenarios.

We selected the models based on their performance on the TRUDI dataset. The first stage employs Mask R-CNN [8] to segment TUs in images. The model is pre-trained on COCO [19] and fine-tuned on the TRUDI dataset only with the *container*, *tank_container*, and *trailer* classes. This stage outputs cropped masks of the TU instances for the text detection stage. The second stage uses a DBNet++ [17] model pre-trained on SynthText [6] and

Table 3: Results for stage 1: instance segmentation. "-C" denotes fine-tuning on Combined dataset. "3C" (Three Classes) indicates that the models are trained to distinguish between `container`, `tank_container` and `trailer` classes, while "1C" (One Class) covers all three types of TUs in one class.

Perspective	Model	AP@0.50:0.95	AP@0.50	AR@0.50:0.95
Aerial	RTMDet (1C)	0.22	0.29	0.44
	Mask R-CNN-Swin (1C)	**0.43**	**0.62**	**0.48**
	RTMDet (3C)	0.34	0.44	0.42
	Mask R-CNN-Swin (3C)	0.38	0.56	0.44
	Mask R-CNN-Swin (3C)-C	0.42	**0.62**	0.47
Ground	RTMDet (1C)	0.27	0.43	0.45
	Mask R-CNN-Swin (1C)	0.40	0.62	0.48
	RTMDet (3C)	0.24	0.37	0.38
	Mask R-CNN-Swin (3C)	0.40	0.61	0.47
	Mask R-CNN-Swin (3C)-C	**0.44**	**0.67**	**0.50**
Combined	RTMDet (1C)	0.35	0.52	**0.50**
	Mask R-CNN-Swin (1C)	0.43	0.64	0.49
	RTMDet (3C)	0.41	0.57	0.49
	Mask R-CNN-Swin (3C)	**0.44**	**0.65**	**0.50**

fine-tuned on TRUDI for text detection within the TU masks, addressing challenges such as varying orientations and environmental conditions including bad condition of TU markings. In the final stage, detected text areas are cropped and fed into the RobustScanner [31], a text recognition model pre-trained on the ICDAR15 dataset [11] and fine-tuned on TRUDI. The model recognises text and extracts the ID code. The output is then verified for ISO6436 compliance and associated with the TU. If the text does not comply, it is flagged for further inspection. The pipeline outputs a file that associates TUs and their respective markings. This association is crucial for port monitoring as it enables real-world localisation of each TU via georeferenced images.

5 Experiments and Results

We conducted experiments using TRUDI and its subsets (aerial, ground, and combined perspectives) to evaluate the three stages of TITUS and establish a baseline. Instance segmentation models were fine-tuned on full images, while text detection and recognition models were trained on cropped TU instances and text areas, respectively. The details of the fine-tuning process, including all hyperparameter configurations, are documented in the repository provided in Footnote 1.

Instance Segmentation: The initial stage aims to provide a mask for each TU individually, using an instance segmentation model. For this stage, we fine-tuned two models that were pre-trained on the COCO [19] dataset: Mask R-CNN with a Swin Transformer [8, 25] backbone and RTMDet [27]. We fine-tuned the instance segmentation models in two settings: Three Classes (3C) setting involved fine-tuning the models to differentiate between three TU classes (`container`, `tank_container`, and `trailer`). The second setting, One Class (1C), consolidated all three types of TU classes into a single class as the ID code associated with each unit provides sufficient information to determine its type, thereby simplifying the classification task.

Table 3 shows the results of the instance segmentation model, evaluated using average

Table 4: Results for stage 2: text detection (@0.5 IoU). "-C" indicates that the model is fine-tuned on Combined dataset but tested on either aerial or ground perspective.

Perspective	Model	Recall	Precision	Hmean
Aerial	DBNet++	0.77	0.61	0.68
	DBNet++-C	**0.79**	**0.68**	**0.73**
	DBNet	0.29	0.30	0.29
	PANet	0.44	0.14	0.21
Ground	DBNet++	0.73	0.63	0.68
	DBNet++-C	**0.79**	**0.69**	**0.74**
	DBNet	0.18	0.44	0.25
	PANet	0.48	0.05	0.09
Combined	DBNet++	**0.79**	**0.69**	**0.74**
	DBNet	0.43	0.55	0.48
	PANet	0.57	0.16	0.25

precision (AP) and average recall (AR) across various intersection over union (IoU) thresholds. Mask R-CNN consistently outperforms RTMDet across all settings, especially for average precision, making it the preferred model for this stage in TITUS. For both aerial and ground perspective, training with the one class setting achieves higher precision and recall. However, when the perspectives are combined, both settings produce comparable results. The models benefit from being exposed to diverse viewpoints during training as the combined perspective results in the highest overall scores. Fine-tuning on both aerial and ground datasets and testing on the ground perspective results in superior performance compared to fine-tuning only on the ground perspective dataset. This suggests that the ground perspective benefits from the inclusion of aerial perspective images during the fine-tuning process. Conversely, fine-tuning on the combined dataset and testing on the aerial dataset shows that the aerial perspective does not gain benefits from the ground perspective images.

Text Detection: For the text detection stage, we fine-tuned three text detection models which were pre-trained on SynthText [6]: PANet [22], a model originally designed for instance segmentation, DBNet [16], a prominent real-time scene-text detection model, DBNet++ [17], an improved version of DBNet with better feature fusion and differentiable binarisation. For fine-tuning this stage, each TU mask was cropped based on the ground truth annotations. These crops were then fed into the model as input for fine-tuning.

Table 4 presents results for text detection with the evaluation metrics recall, precision and harmonic mean (Hmean) for detection models. DBNet++ achieves the best performance across all three settings, which makes it the preferred text detection model for the second stage of TITUS. Training on the combined dataset results in the highest Hmean for all three models, suggesting that multi-perspective training increases model robustness and generalisation for text detection. Fine-tuning on the combined dataset and testing on either the aerial or ground perspective results in improved performance compared to fine-tuning solely on the respective perspective's dataset. This indicates that both aerial and ground perspectives benefit from each other during the fine-tuning process. This is not observed in instance segmentation, since TUs are 3D objects that can appear visually different from different perspectives, while text can be considered as a 2D object that appears similar across perspectives.

Text Recognition: For the final stage, we fine-tuned three text recognition models, which were initially pre-trained on the ICDAR15 dataset [11]: SVTR [2], a model that uses transformer encoder-decoder architecture, RobustScanner [41], a model that employs a feature

Table 5: Results for stage 3: text recognition. Since the ID codes are always uppercase and do not include symbols, the case and symbols are ignored. "-C" indicates that the model is fine-tuned on the Combined dataset but tested on either aerial or ground perspective.

Perspective	Model	Word acc.	Char. recall	Char. precision
Aerial	SVTR	0.60	0.86	**0.90**
	RobustScanner	0.64	0.87	0.87
	RobustScanner-C	**0.68**	**0.88**	0.88
	SAR	0.10	0.18	0.36
Ground	SVTR	0.49	0.69	**0.82**
	RobustScanner	0.50	0.70	0.72
	RobustScanner-C	**0.54**	**0.73**	0.73
	SAR	0.05	0.21	0.33
Combined	SVTR	0.57	0.81	**0.90**
	RobustScanner	**0.63**	**0.83**	0.84
	SAR	0.13	0.39	0.43

Table 6: End-to-end evaluation results of TITUS across aerial, ground, and combined perspectives, showing precision, recall, F1 score, and accuracy.

Perspective	Precision	Recall	F1 Score	Accuracy
Aerial	**0.45**	**0.30**	**0.36**	**0.22**
Ground	0.25	0.19	0.22	0.12
Combined	0.39	0.27	0.32	0.19

fusion model and suitable for contextless text recognition like ISO6346 compliant ID code, and SAR [32], an early model using a 2D attention mechanism. During the fine-tuning, the model's input were individually cropped text areas based on the ground truth annotations.

Table 5 shows the results of the three text recognition models that have been used for automatic container code recognition. These models are evaluated on character-level performance (recall and precision) and word-level accuracy. In this context, a word could be the whole ISO6346 compliant ID code, size and type code, or for some trailers the registration plate number written on the TUs. RobustScanner achieves the best balance of high word accuracy and character-level performance across all perspectives. SVTR has a similar though slightly lower performance compared to RobustScanner in word accuracy. SAR, on the other hand, performs significantly worse than the other two, indicating difficulty to recognise container or trailer ID codes regardless of the perspective. Similar to the text detection stage, fine-tuning on the combined dataset results in better performance in all perspectives.

Compared to the aerial perspective, the three text recognition models underperform when dealing with the ground perspective. Unlike UAVs, ground vehicles do not stop to capture images, resulting in increased motion blur which hinders the performance despite the potentially shorter distance between the camera and the text. Additionally, vertical codes, common on the sides of TUs, are more prominent in the ground perspective. Occlusion is also more common than in aerial images due to the presence of other objects or people. Because of these reasons, the ground perspective is the most challenging perspective in TRUDI.

End-to-End Evaluation: Using the best performing models on the combined set, we evaluated TITUS to assess its end-to-end identification performance. The results in Table 6 highlight benchmarking capabilities of TRUDI and the challenges it offers. Despite the complexity of real-world data, the proposed pipeline is capable of effectively identifying TUs.

In practical deployments, mobile cameras are expected to capture video streams rather than static images, offering temporal continuity that can significantly enhance identification. This continuous input give the system with multiple opportunities to observe a TU in consecutive frames, thereby increasing the likelihood of correct recognition and association, even under challenging conditions.

6 Conclusion

The TRUDI dataset, comprising 35,034 instances of five classes, addresses the need for a publicly available benchmark dataset for the TU identification task. This dataset encompasses multiple perspectives and real-world operational conditions, including images captured under various lighting and weather conditions. This allows for a more comprehensive evaluation and ensures that models trained on TRUDI can handle real-world scenarios effectively. The proposed TITUS pipeline follows a three-stage approach including 1) TU segmentation, 2) ID text detection, and 3) text recognition. It offers a robust and flexible solution for TU identification, particularly suitable for mobile cameras mounted on aerial or ground vehicles. The evaluation of the pipeline on the TRUDI dataset demonstrates its effectiveness for TU identification. These results set a strong baseline for future research for each individual processing step and the entire TU identification pipeline. The contributions of TRUDI and TITUS are expected to facilitate the development of new applications and methods in TU identification, enhance benchmarking, and improve operational efficiency in multipurpose port logistics.

Acknowledgements

The project is supported by the German Federal Ministry for Digital and Transport (BMDV) in the funding program Innovative Hafentechnologien II (IHATEC II).

References

[1] Alejandro Diaz-Diaz, Franciso Parrilla, R De La Iglesia, Rafael Barea, and Luis M Bergasa. Wagon and container codes detection and recognition based on yolov8. In *2024 7th Iberian Robotics Conference (ROBOT)*, pages 1–6. IEEE, 2024.

[2] Y Du, Z Chen, C Jia, X Yin, T Zheng, C Li, Y Du, and YG Jiang. SVTR: Scene text recognition with a single visual model. arxiv 2022. *arXiv preprint arXiv:2205.00159*, 2022.

[3] Glenn Jocher et. al. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support, October 2021. URL https://doi.org/10.5281/zenodo.5563715.

[4] Int'l Organization for Standardization. Freight containers — Coding, identification and marking. Standard, Geneva, CH, 2022.

[5] Philippe Gaussier, Sorin Moga, Mathias Quoy, and Jean-Paul Banquet. From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 12(7-8):701–727, 1998.

[6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] Nong Thi Hoa and Nguyen Van Tao. Build an effective system for container code recognition. In *International Conference on Advances in Information and Communication Technology*, pages 32–39. Springer, 2023.

[10] Chung-Chian Hsu, Yu-Zen Yang, Arthur Chang, SM Salahuddin Morsalin, Guan-Ting Shen, and Li-Shin Shiu. Automatic recognition of container serial code. In *2023 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, pages 257–258. IEEE, 2023.

[11] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[12] Ray Khuboni, Sanele Hlabisa, and Jules R Tapamo. Confidence-guided shipping container code recognition using deep learning. *Available at SSRN 5021624*, 2025.

[13] Brett Koonce. MobileNetV3. In *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*, pages 125–144. Springer, 2021.

[14] John CM Lee. Automatic character recognition for moving and stationary vehicles and containers in real-life images. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 4, pages 2824–2828. IEEE, 1999.

[15] Yanchao Li, Hao Li, and Guangwei Gao. Towards end-to-end container code recognition. *Multimedia Tools and Applications*, 81(11):15901–15918, 2022.

[16] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proc. of the AAAI Conf. on Artificial Intelligence*, volume 34, pages 11474–11481, 2020.

[17] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):919–931, 2022.

[18] Han Lin, Peng Yang, and Fanlong Zhang. Review of scene text detection and recognition. *Archives of computational methods in engineering*, 27(2):433–454, 2020.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence 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*, pages 740–755. Springer, 2014.

[20] Zhehao Lin, Chen Dong, and Yuxuan Wan. Research on intelligent recognition algorithm of container numbers in ports based on deep learning. In *Int'l Conf. on Intelligent Computing*, pages 184–196. Springer, 2024.

[21] Shiran Liu, Zhaoqiang Guo, Yanhui Li, Chuanqi Wang, Lin Chen, Zhongbin Sun, Yuming Zhou, and Baowen Xu. Inconsistent defect labels: Essence, causes, and influence. *IEEE Transactions on Software Engineering*, 49(2):586–610, 2022.

[22] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[23] Tao Liu, Xianqing Wu, and Fang Li. Lightweight container number recognition based on deep learning. *International Journal of System Assurance Engineering and Management*, pages 1–14, 2025.

[24] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9809–9818, 2020.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[26] Ho C Lui, Chung M Lee, and Fang Gao. Neural network application to container number recognition. In *Proceedings., Fourteenth Annual International Computer Software and Applications Conference*, pages 190–195. IEEE, 1990.

[27] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. RTMDet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022.

[28] Duncan McFarlane and Yossi Sheffi. The impact of automatic identification on supply chain operations. *The International Journal of Logistics Management*, 2003.

[29] Hoang-Sy Nguyen, Cong-Danh Huynh, and Nhat-Quan Bui. Digital transformation for shipping container terminals using automated container code recognition. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 21(3):535–544, 2023.

[30] Sayali Nimkar, Sanal Varghese, and Sucheta Shrivastava. Contrast enhancement and brightness preservation using multi-decomposition histogram equalization. *arXiv preprint arXiv:1307.3054*, 2013.

[31] Wei Pan, Yangsheng Wang, and Hongji Yang. Robust container code recognition system. In *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, volume 5, pages 4061–4065. IEEE, 2004.

[32] Jifang Pei, Yulin Huang, Weibo Huo, Yin Zhang, Jianyu Yang, and Tat-Soon Yeo. SAR automatic target recognition based on multiview deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2196–2210, 2017.

[33] Ravindra Shetty, Rebeca Cáceres, John Pastrana, and Luis Rabelo. Optical container code recognition and its impact on the maritime supply chain. In *Proceedings of the 2012 Industrial and Systems Engineering Research Conference*, pages 1535–1544, 2012.

[34] Xiaoning Shi, Dongkai Tao, and Stefan Voß. RFID technology and its application to port-based container logistics. *JOCEC*, 21(4):332–347, 2011.

[35] Ray Smith. An overview of the Tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[36] Jana Teegen, André Kelm, Ole Grasse, Maris Hillemann, Emre Gülsoylu, and Simone Frintrop. Drone-based identification of containers and semi-trailers in inland ports. EasyChair Preprint 14025, EasyChair, 2024.

[37] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. MobileOne: An improved one millisecond mobile backbone. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7907–7917, 2023.

[38] Dapeng Yang, Guanghui Wang, Mingtang Liu, Shuang Yue, Hao Zhang, Xiaokang Chen, and Mengxiao Zhang. Lightweight container code recognition based on multi-reuse feature fusion and multi-branch structure merger. *Journal of Real-Time Image Processing*, 20(6):108, 2023.

[39] Li Yao, Chenchen Tang, and Yan Wan. Advanced text detection of container numbers via dual-branch adaptive multi-scale network. *Applied Sciences*, 15(3):1492, 2025.

[40] Meng Yu, Shanglei Zhu, Bao Lu, Qiang Chen, and Tengfei Wang. A two-stage automatic container code recognition method considering environmental interference. *Applied Sciences*, 14(11):4779, 2024.

[41] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. RobustScanner: Dynamically enhancing positional clues for robust text recognition. In *European conference on computer vision*, pages 135–151. Springer, 2020.

[42] Ran Zhang, Zhila Bahrami, and Zheng Liu. A vertical text spotting model for trailer and container codes. *IEEE Transactions on Instrumentation and Measurement*, 70: 1–13, 2021.

[43] Jian Zhao, Ning Jia, Xianhui Liu, Gang Wang, and Weidong Zhao. A practical unified network for localization and recognition of arbitrary-oriented container code and type. *IEEE Trans. on Instrumentation and Measurement*, 2024.

[44] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[45] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.