# RAILCAR DETECTION, IDENTIFICATION AND TRACKING FOR RAIL YARD MANAGEMENT

*Ming-Ching Chang*[1], *Guangliang Zhao*[2], *Abhineet Kumar Pandey*[1], *Andrew Pulver*[1], *Peter Tu*[2]

[1] University at Albany, State University of New York, NY, USA
[2] GE Global Researh Center, NY, USA

## ABSTRACT

We present a video analytics system combining railcar detection, classification, Federal Railroad Admin. (FRA) text identification, and logo detection into a system for locomotive transportation and yard management. Existing RFID-based systems are limited by sensor deployment and cannot visually identify railcars when they are away. As there are typically tens of tracks and hundreds of railcars in a yard, an automatic vision system is desirable. The proposed AI system is developed for autonomous yard inventory checking, such that the arrival, departure, and movement of individual railcars can be automatically monitored and managed in the facility. Our system consists of multiple cameras with edge computing devices installed at check points (track entrances and branches), such that visual detection and tracking of railcars can be performed and meta-data can be exchanged. After knowing the railcar locations and types, scene text detection is performed to search and recognize FRA ID markings and logos that can uniquely identify each railcar. Information fusion a database in the central hub can further improve railcar identification and reduce errors. Early results on real-world field collected data demonstrate the efficacy of the proposed approach.

**Keywords:** railcar, detection, tracking, scene text detection, OCR, Federal Railroad Administration, FRA, locomotive transportation, yard management, edge computing.

## 1. INTRODUCTION

With the raise of AI visual analytics, deep neural network (DNN) based vision systems can now achieve high technical readiness for industrial use in logistics and asset management [2]. In this paper, we present a video system for locomotive yard inventory management applications (Fig.1), by integrating multiple modules including railcar detection, classification, Federal Railroad Administration (FRA) text identification, logo detection, message passing, and control center database logger. The development is related to recent advancements in autonomous driving cars [16], car license plate recognition (LPR) [1], and smart transportation systems [10], however the use case and problem scenarios for railcar applications are different, as discussed in the following.

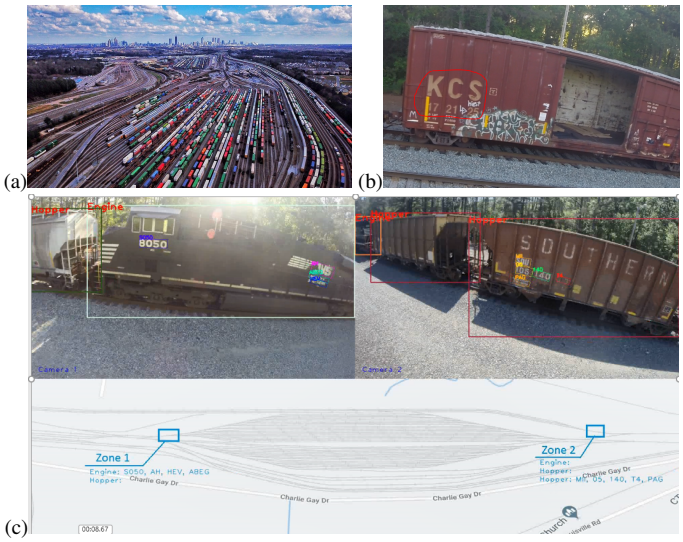


**Fig. 1**. **Overview.** (a) A busy rail yard close to Atlanta. (b) Graffiti on a boxcar makes the FRA texts hard to recognize. (c) Yard layout and camera deployment in our experimental setup. Top two views show railcar detections, FRA text identifications, and logo detections from the two edge devices.

(a) from www.reddit.com/r/Atlanta/comments/8099iu/from_csx_terminal_west_to_the_city/

Todays locomotive yard uses RFID tag readers at yard entrance to detect train come/leave, by reading each RFID tag attached on individual railcars. RFID tags may be damaged or missing, and causing missed detections. In a yard, there are typically tens of tracks and hundreds of railcars awaiting sorting or switching, where the *'switching operations'* (assembly/disassembly of railcars) optimize for the least times of coupling operations and less distance traveled, to achieve sooner new configurations for the next outbound train. In practice, RFID tag reading stations are only available at track entrances or branches, since it is not practical to install RFID reader at every track. Nowadays, yard rolling stock switch and inventory management still rely on manual processes, which is tedious, time consuming and error prone.

In US, FRA regulations enforce that operating railcars on railroads must be labeled with a Standard Carrier Alpha Code (SCAC), which consists of two to four letters. In practice, each railcar is typically labeled with a unique ID consisting

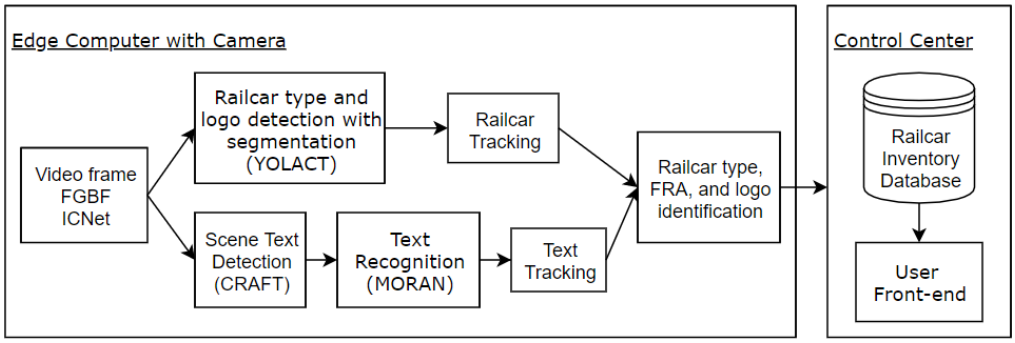


**Fig. 2**. **System architecture and workflow** for automatic railcar detection, FRA ID identification and tracking for locomotive yard management applications.

**Fig. 3**. **Rail track localization** using semantic segmentation.

of SCAC and/or 1-6 digits, the **FRA ID**, that uniquely identifies each railcar (an analogy to the license plate for street vehicles). A major goal of this study is to detect and identify such FRA ID for each rail car. Typical rail car types include *tank*, *hopper*, *boxcar*, *gondola*, *flatcar*, *etc.* (Fig.4). FRA ID may appear in different font type, size, at different locations and layouts there-in. Graffiti, paint peeling off, or wearings can cause challenges for the visual detection and recognition of FRA texts.

We develop an autonomous visual railcar detection, identification, and tracking system for yard management, with an aim to automatically identify the arrivals, departures, and locations of railcars in the yard. Our vision system consists of one or more edge devices deployed at check points (yard entrances or track branches) and a control center hub (Fig.1c). The RGB video input are fed into two branches of pipeline. and exits of the yard. Workflow at each edge device consists of two branches of pipeline (Fig.2): the first branch performs rail car detection, logo detection and tracking, and the second deals with scene text detection and recognition for FRA text identification. State-of-the-art AI computer vision techniques including YOLACT [5] object detection and segmentation, and CRAFT [4] scene text detection, MORAN [8] text recognition, among the others [11, 15, 17, 12, 13, 3] are evaluated, adopted and integrated into the pipeline.

In this study, we constrain the FRA ID search space to be a known, limited set of railcars that are scheduled to visit the targeted yard. This problem setup aligns with business usage, as the rail yard control center typically has access to customer train schedules. With such known anticipated railcars that will appear, our video analytics results *i.e.* recognized railcar texts can be matched against the scheduled railcar lists to refine toward the final FRA ID lists. Experiments are performed on a railcar dataset collected from a real-world locomotive yard in the US. Evaluations of railcar FRA identification precision/recall (PR) are performed on this dataset in § 4. Results show that our method can achieve accuracy of $80.54\%$. To the best of our knowledge, this is the first video analytics system of the kind for rail car management applications.

## 2. BACKGROUND

**Railcar inventory management.** Traditionally railway industry relies on Radio Frequency Identification (RFID) for rail car logistic management [9]. With the rise of AI technologies, video analytic systems now have growing impact for the next-generation rail yard inventory control.

**Visual object detection** has been studied extensively. Two-stage detectors such as Faster-RCNN and Mask-RCNN [6] are generally more accurate but slower. Single-stage detectors such as SSD are faster but less accurate. YOLOv3 [11] is a popular real-time object detection model based on improvements on previous YOLO generations. YOLACT [5] is a simple fully-convolutional model that can achieve real-time instance segmentation, with a design of prototype mask set generation and per-instance mask coefficient prediction. Object detection with high-quality instance mask can be obtained by linearly combining the mask prototypes with the coefficients, without the dependency of repooling.

**Multiple object tracking** approaches typically rely on the tracking-by-detection paradigm. Given per-frame detection results in bounding boxes, visual tracker performs matching (based on appearance similarity or geometric consistency, tracklet association and update. Popular methods include Kalman filtering, DeepSORT [15], or data-driven tracking methods such as the TrackletNet [14].

**Scene text detection and recognition.** Recent advances can be organized into three broad categories (see survey [18]): (1) *scene text detection* [17, 12, 4], concerning the discovery and localize texts from natural images, (2) *scene text recognition* [13, 3, 8], focusing on understanding the texts from the detected character regions, and (3) *end-to-end scene text spotting* using a single network such as [7].

**Scene text detection.** SegLink [12] uses a FCN that decomposes scene texts into locally detectable segments (oriented box covering a part of a word) and links (which connect adjacent segments). Recently, CRAFT [4] can effectively detect text area by exploring each character with region awareness and modeling the affinity between characters, without the need of individual character level annotations.

**Scene text recognition (STR).** CRNN [13] is among the

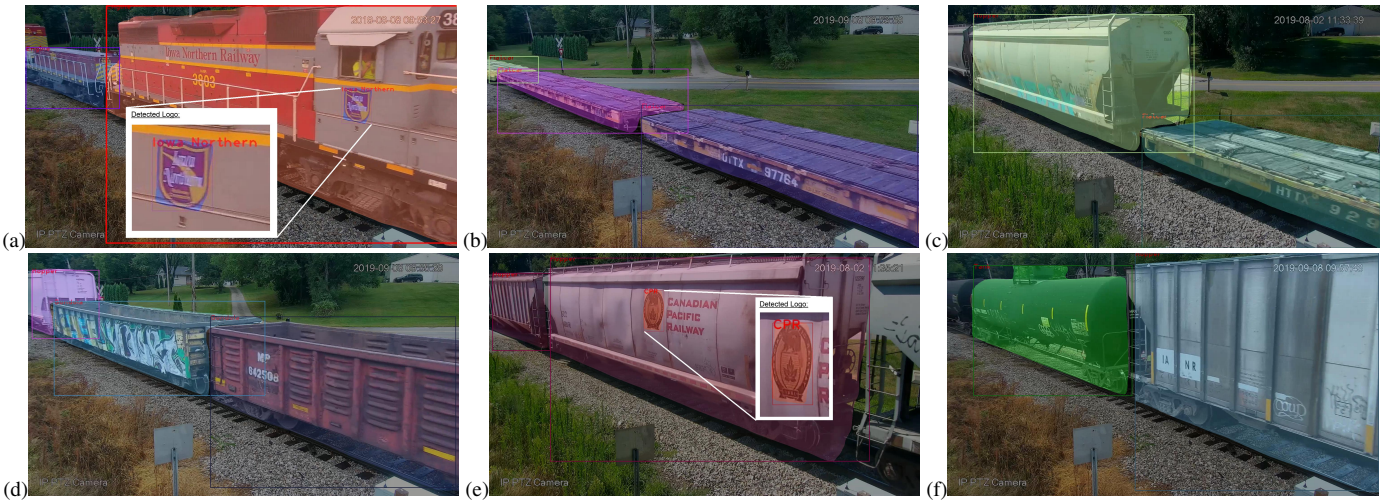


**Fig. 4**. YOLACT railcar and logo detection (type classification) with instance segmentation results. (a) Detected 2 engines and 1 logo. (b) Detected 3 flatcars. Note that the leftmost is only partially observed but can still be detected. (c) Detected 1 hopper and 1 flatcar. (d) Detected 1 hopper and 2 gondola cars. (e) Detected 2 hoppers and 1 logo. (f) Detected 1 tank and 1 hopper.

earliest end-to-end trainable methods for image-based sequence recognition. MORAN [8] consists of a multi-object rectification network and an attention-based sequence recognition network, which can effectively recognize irregular scene texts. The work of [3] introduce a unified four-stage framework that allows for consistent training and evaluation of STR methods. Google Tesseract designed for document OCR [1]is not suitable for scene text recognition.

## 3. METHOD

The proposed video analytics system consists of a set of fixed RGB cameras at the both ends of the yard as in Fig.1c. As the train enters the yard they can be detected using standard foreground/background (FGBG) analysis. We use OpenCV to detect any train passing. We further combine this train detection approach with semantic segmentation (Fig.3). This effectively narrows down the ROI to speed up analysis. This way the train-passing detection can run on long hours of videos, and determine if it is necessary to continue the pipeline.

For the video frames with train(s), further pipeline of two branches in Fig.2 are carried out and then a fusion is performed to uniquely identify each railcar: (1) The **railcar branch** performs per-car detection, segmentation, logo detection/tracking (if found) and railcar tracking. (2) The **text branch** performs scene text detection and recognition. (3) The **FRA fusion module** then take results from both branches to robustly determine the FRA ID that uniquely identify each railcar. (4) Per-frame results generated at each edge device are sent to the **control center** via message passing for database logging, inventory update, and user front-end. Note the two branches can be run in parallel *asynchronously* to maximize the processing frame rate at the edge device.

[1] https://github.com/tesseract-ocr/tesseract

### 3.1. Railcar and logo detection, segmentation, tracking

The railcar branch pipeline in Fig.2 start with fast YOLACT [5] detection that can localize 11 types of railcars and 15 logo classes. Logo detection are useful in joint improvement of railcar identification when combined with recognized FRA texts. The YOLACT detection results come with instance segmentation (Fig.4), which can be used to estimate a rectified mask via affine transformation that can normalize the localization of FRA ID texts within the railcar mask. Note that partially appearing railcars are properly annotated in our training set, such that our YOLACT model can detect partly observed railcars as in Fig.4b,d. This is a major difference compared to standard street car detection methods (*e.g.* Mask-RCNN [6]), as railcars are much longer in shape. There is a trade-off between setting a wider camera view in order to capture the full railcar, or a narrower view to assure enough pixel resolution for FRA text detection.

We also experimented other object detectors including YOLOv3 [11]. However the 2D detection boxes is less useful compared to the segmentation masks obtained from YOLACT, which can help localizing FRA texts and filtering out possible graffitis or unwanted texts after the rail car type is known.

**Tracking.** All detected railcars and logos are fed into a DeepSORT [15] tracker to construct respective tracklets for robustly tracking and counting. Logo(s) inside a railcar mask associated with it for identification.

### 3.2. FRA text detection and tracking

The text branch pipeline in Fig.2 adopts CRAFT [4] scene text detection followed by MORAN [8] text recognition for FRA ID determination. We have also investigated other scene text detection [17, 12] and recognition [13, 3] approaches, and found the aforementioned pipeline is most effective.

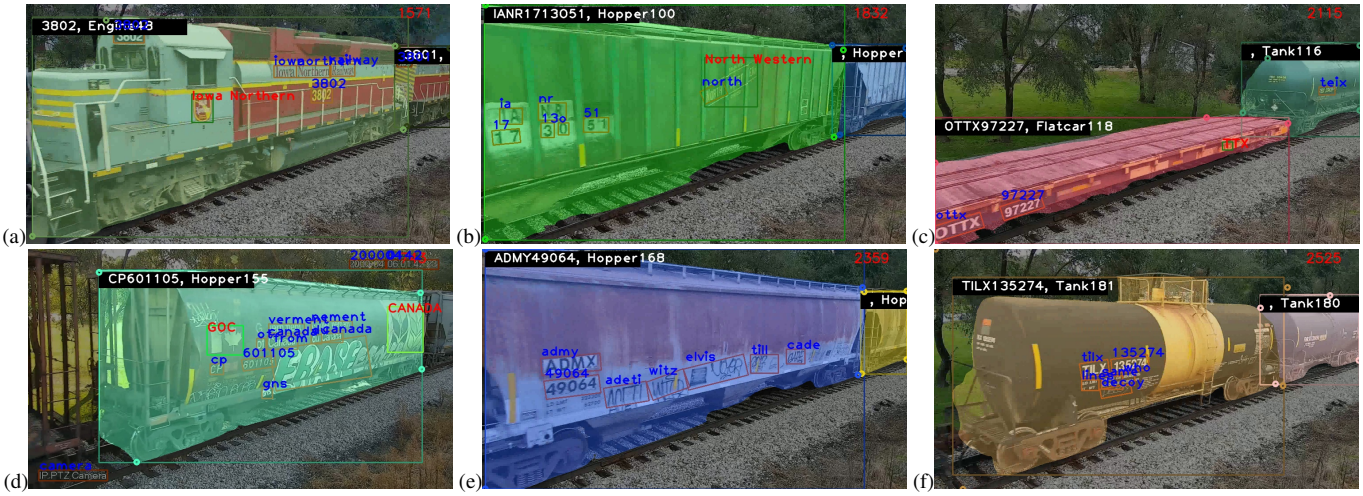


**Fig. 5**. **Visual results.** Recognized FRA ID with railcar type are shown on upper-left. In several cases the noisy CRAFT text detection and MORAN recognition results are successfully filtered out using the FRA determination rules described in §3.3.

### 3.3. FRA ID text identification

The FRA ID consists of 6 to 8 characters or digits in one or two rows. In some case it contains only 6 digits in a single row, and in other cases there are two rows with 4 characters and 6 digits. FRA text identification can be challenging due to character fade out, graffitis, clutter backgrounds, blurring from fast moving training, and environmental variations.

We perform tracking of each individual character across frame for robust aggregation, and then filter texts using a ROI calculated from YOLACT railcar segmentation mask. For example, as in Fig.5, for gondola or box car the FRA texts can only appear at the left side of the railcar body. Detected scene texts are grouped and associated according to the FRA ID string formats, and in several cases OCR errors can be recovered. For example, for an ID with 5 digits and 1 character, the character is likely incorrect (such as 'I' vs '1', 'B' vs '8').

The next step is the re-assembly of the FRA ID via grouping texts. Challenges here include the mix of ID texts with other texts printed on the car body as in Fig.5d,e,f. The spacing between individual characters can vary, and the same FRA ID can appear at multiple locations of the car. All text candidates are properly filtered, assembled, and organized to match the FRA IDs from a known list of anticipating cars. Texts that are far off, too large or too small are ruled out.

### 3.4. Control center and database

The identified railcars with FRA IDs, types and logos at each edge device are sent to the control center using message passing for database logging and user front-end. The database stores detected train and railcar information for inventory queries. Railcar identification information are aggregated, and until the train moves completely out of the view, we then determine the FRA ID based on majority voting from the tracklet. One can further consider the matching order of railcars to refine identification results.

## 4. EXPERIMENTAL RESULTS

We have collected a **field dataset** using Dahua 59430UNI and 6CE230UNI wide-angle PTZ cameras set at heights ranging from 4 to 12 feet, at distances ranging from 10 to 30 feet to the rail tracks. Videos from day and night and in various weather conditions are collected. We selected $6,112$ frames for dense annotation, which includes $26,899$ railcars of 11 types, $4,003$ logos in 14 types, and $8,921$ FRA IDs. The YOLACT dataset is split into 70% for training, 15% validation, and 15% testing sets. The YOLACT, CRAFT, and MORAN models are implemented in PyTorch.

The end-to-end speed of our system (including video capturing and processing) running on NVIDIA AGX XAVIER varies from 4 to 7 frames per second. Speed bottleneck is at the FRA text detection module, since some cars such as Fig.5f contain many texts unrelated to FRA however our pipeline has to detect them before they can be ruled out as in § 3.3.

**Results.** We compare the detected FRA ID against manual groundtruth labeling for two sets of test video containing about 50 railcars. Only cases where the whole detected FRA ID string match the groundtruth are considered true positives. We obtain Precision-recall (PR) accuracy of $80.54\%$. Note the intermediate YOLACT car type and logo detection accuracy are close to $95\%$. FRA text detection and recognition is relatively more difficult.

## 5. CONCLUSIONS

We present a video analytics system combining railcar detection, type classification, FRA ID text identification, and logo detection into a system for locomotive transportation and yard management. **Future work** includes thorough real-time, online evaluation and extension with image re-identification for site-wide asset tracking.

## 6. REFERENCES


[1] C.-N. E. Anagnostopoulos, I. E. Anagnostopolous, I. D. Psoroulas, V. Loumos, and E. Kayafas. License plate recognition from still images and video sequences: A survey. *IEEE Trans. Intell. Transp. Syst.*, 9(3):377–391, 2008. 1

[2] G. Ananthanarayanan, V. Bahl, P. Bodk, K. Chintalapudi, M. Philipose, L. R. Sivalingam, and S. Sinha. Real-time video analytics the killer app for edge computing. *Computer*, 50(10):58–67, 2017. 1

[3] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, 2019. 2, 3

[4] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019. 2, 3

[5] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. YOLACT: Real-time instance segmentation. In *ICCV*, October 2019. 2, 3

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 2, 3

[7] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. FOTS: Fast oriented text spotting with a unified network. In *CVPR*, 2018. 2

[8] C. Luo, L. Jin, and Z. Sun. MORAN: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019. 2, 3

[9] B. Malakar and B. K. Roy. Survey of RFID applications in railway industry. In *ACES*, pages 1–6, 2014. 2

[10] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P. Chakraborty, T. Huang, J.-H. Hwang, and S. Lyu. The 2019 AI city challenge. In *CVPR Workshop*, 2019. 1

[11] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv:1804.02767*, 2018. 2, 3

[12] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, pages 2550–2558, 2017. 2, 3

[13] 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. *PAMI*, 39, 2016. 2, 3

[14] G. Wang, Y. Wang, H. Zhang, R. Gu, and J. Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. *CoRR*, abs/1811.07258, 2018. 2

[15] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 2, 3

[16] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda. A survey of autonomous driving: Common practices and emerging technologies. In *preprint arXiv:1906.05113*, 201. 1

[17] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. EAST: An efficient and accurate scene text detector. In *CVPR*, 2017. 2, 3

[18] Y. Zhu, C. Yao, and X. Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers Comput. Sci.*, 10(1):19–36, 2016. 2