Moving train wheel axles automated detection, counting, and tracking by combining AI with Kalman filter applied to thermal infrared image sequences

Boualem Merainani, Thibaud Toullier, Jean Dumoulin

Moving train wheel axles automated detection, counting and tracking by combining AI with Kalman filter applied to thermal infrared image sequences

Boualem Merainani[a], Thibaud Toullier[a], and Jean Dumoulin[a]

[a]Université Gustave Eiffel, Inria, COSYS-SII, I4S Team, F-44344 Bouguenais, France

ABSTRACT

Hot boxes, which refer to overheated rail-road car wheels and bearings, pose a significant threat to railway operations. Failure to detect and address hot boxes promptly can lead to catastrophic accidents such as derailments and fires. Current way-side hot box detectors operate on the principle that an axle bearing will emit a large amount of heat when it is close to failing. They require principally an infrared (IR) sensor mounted at specific locations along the track, and a signal source coming from a wayside detectors or track circuits to detect if a train is approaching. The IR sensors scanning location, however, should be carefully selected to avoid under/over predicting the operating temperature of the axle bearings and wheels. The dependency of a signal source to activate the system may be problematic as well, not to mention its implementation and maintenance costs. The main contribution of this paper lies with the development of an automatic hot box detection, tracking and counting method by only using the IR cameras. The method combines the YOLO algorithm with the Kalman filter as a tracker. The method was tested with original datasets built with IR images taken from two wayside camera models, cooled and uncooled cameras. The experiments have been conducted on both freight and passenger trains at different times of the day, under clear weather conditions. Apart from the promising results obtained by YOLO, it is found that the Kalman filter further improves the tracking and thus the detection performance, minimizing thereby the incorrect detection or missed detection.

Keywords: Thermal Infrared Imaging, Deep learning, YOLO, Kalman filter, Hot boxes, Automated detection

1. INTRODUCTION

The importance of rail transportation cannot be understated, as it plays a critical role in both the global economy and our daily lives. However, as transportation demands grow, infrastructure managers face significant challenges in developing efficient, reliable, and sustainable solutions that prioritize safety and operational performance.

Rolling stock relies on several key components to operate effectively, with the bogies, suspension and brake systems being crucial elements. The wheels and axle bearings are even more important. They support the weight of the train, provide traction, and ensure smooth and safe movement along the track. Furthermore, they are subject to extreme stresses and loads but, at the same time, are exposed to environmental contaminants. Damage may occur and the axle bearings may overheat under these circumstances. The wheels may experience slide and become locked, resulting in overheating and causing thermal stress damage as well as a flat spot. If these damages are not detected and addressed in time, they can result in disastrous incidents like train derailments and fires. Remote and contactless condition monitoring technologies have emerged as a promising solution to mitigate such risks. Some of them rely on acoustic emissions (AE), such as railway bearing acoustic monitoring (RailBAM), while others use thermal sensors, like hot box detector (HBD), hot wheel detector (HWD) and infrared cameras-based thermal imagery. AE-based systems, however, face the issue of signal attenuation, meaning that the AE sensor needs to be in close proximity to the source of the signal.[1] Unfortunately, this is not always feasible. The HBD and HWD systems have been implemented in several railway networks in many countries, demonstrating their capacity and potential. However, these systems have limited effectiveness due to several reasons. Firstly, the installation and maintenance costs is high, which can be a significant barrier to their deployment, particularly

Further author information: (Send correspondence to J.D.)
J.D.: E-mail: jean.dumoulin@univ-eiffel.fr

over short distances. Second, precise temperature measurements rely on accurate calibration of the scanning zone of these systems.[2] Moreover, their operation is contingent upon the availability of other facilities, such as triggers that are responsible for activating and deactivating the system.

Recent improvements in image sensor technology and processing capabilities have made machine vision-based techniques a potential option for detecting hot wheels and hot axle bearings in an efficient and cost-effective manner. Deilamsalehy et al.[3] used wayside uncooled thermal infrared cameras to detect sliding wheels. The wheels were detected using Hough transform (HT) and the classification was done using support vector machine on the basis of the histogram of oriented gradients features. Toullier et al.[4] developed an approach to detect hot boxes in IR images taken by two wayside camera models. The approach rely on convolution with disk-shaped structuring elements. Then, a voting map is generated, using HT, to represent the probability of having a hot box at a given pixel. However, the aforementioned studies suffer from low detection accuracy which considerably limits their application. The authors, in,[5] proposed an approach for undercarriage inspection using machine vision analysis, which involved generating panoramic images from videos captured by a combination of multi-spectral imaging camera and a IR camera. Min-Soo[6] used an infrared camera to monitor the underbody components of railway cars and used the cross-correlation to detect abnormal heat. However, interrupting traffic flow for system installation and maintenance is not easy to manage.

Combining the strengths of thermal cameras and deep learning techniques may hold tremendous promise for improving hot box identification accuracy with a possible expansion to tracking and temperature measurement. Object detection models are usually classified into two families: one-stage and two-stage. The two-stage family include region based convolutional neural networks (R-CNN),[7] fast R-CNN,[8] faster R-CNN.[9,10] In turn, single shot multi-box detector (SSD)[11,12] and you only look once (YOLO)[13–15] are related to one-stage family detectors. YOLO-v4,[16] is considered one of the most outstanding representatives of the YOLO series due to its performance in terms of high robustness and fast recognition. However, all object detection algorithms are prone to certain limitations and challenges. They only see one frame a time. Thus, when objects are in motion, such as train hot axles, time-dependent context is lost. They are prone to miss predictions as well. This problem exacerbates when dealing with IR images, as they often exhibit weak textures and lack of fine details and distinct features observed in visible light images. Nevertheless, the Kalman filter can provide a potential solution to mitigate these limitations. It can be used to bridge the gap between frames, thereby enhancing the detection accuracy. The Kalman filter constructs a mathematical model with a state consisting of position and velocity for every axle. Besides, by taking into account the straight-line trajectory of the train axles, the first predicted parameter, in particular, will help us supporting or rejecting the detection result obtained from detector.

This paper introduces a comprehensive approach for detecting, counting, and tracking of moving train wheel axles in IRTS. The approach combines the power of the YOLO algorithm for object detection and the Kalman filter for accurate tracking. To thoroughly assess the performance of our approach, two different IR camera models were chosen: a cooled camera and an uncooled camera. This selection was motivated by the disparities in image quality they offer and their cost. The remainder of this paper is organized as follows. Section 2, presents the experiments conducted on a rail line open to traffic and describes the hot box detection and tracking system, including YOLO and the Kalman tracking algorithm. The result and discussion will be shown in Section 3. Finally, Section 4 concludes the paper.

2. EXPERIMENTAL DESIGN AND PROCEDURES

2.1 Data description

The study involved conducting in-situ experiments on a rail line that experiences a regular traffic from freight and passenger trains. The data collection phase involved the use of two IR camera types, cooled camera (mounted with SWIR FPA cooled detector) and uncooled camera (mounted with a LWIR FPA uncooled microbolometer detector), capturing images at various times of the day under clear weather conditions. The cooled camera sensor was tuned to capture thermal images with a spatial resolution of 448×239 pixels at $250Hz$, while the uncooled one was tuned to capture thermal images with a higher resolution, 640×240, but at lower frame rate of $100Hz$.

Throughout the experiments, we recorded the passage of more than 100 trains. From the collection, a subset of 874 IR images, taken by the cooled camera, and 908 images by the uncooled camera, were carefully

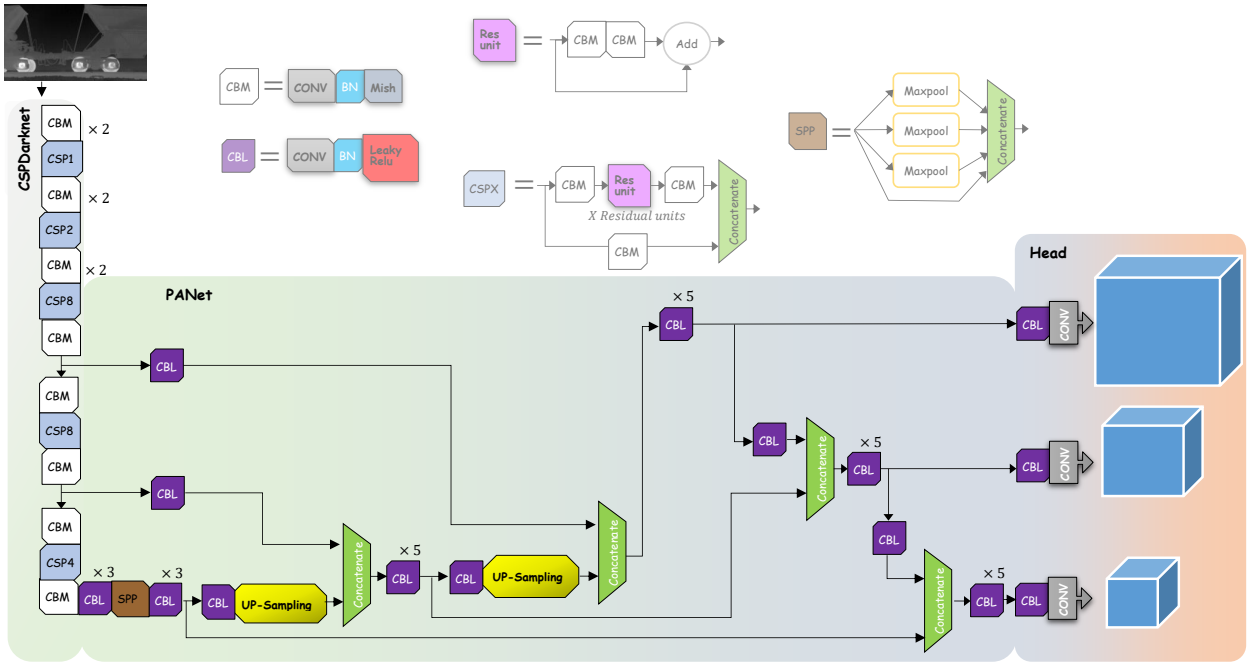

Figure 1: Structure of YOLO

selected for training purposes, ensuring the inclusion of diverse and representative samples. Furthermore, we have incorporated data augmentation techniques to enrich our dataset by applying both photometric and geometric distortions to the selected data. This training strategy, as demonstrated by Zhang et al.,[17] has been proven effective in improving the overall accuracy of an object detection model.

2.2 Hot box detection using YOLO

YOLO is an object detector that uses a single pass to detect the potential regions in the image where certain objects are present and to classify those regions into object classes. The structural architecture of YOLO-v4 consists of a backbone, a neck and the head. The model backbone is used to extract essential features from a given input image and it is designed based on the Cross Stage Partial Network (CSPNet).[18] The model neck is mainly used to collect feature maps from different stages of the model backbone for generating feature pyramids. YOLO-V4 adopts the Path Aggregation Network (PANet)[19] for the model neck. Finally, the model head is used to perform as the final object detection part of YOLO-V4.

During YOLO training, the calculation of the loss function plays a crucial role as it represents the learning process of the model. The primary objective of YOLO, given an IR image, is to provide the coordinates of bounding boxes and the corresponding class label of the detected objects. To achieve this, the model divides the input IR image into $S \times S$ grids cells, where each grid generates B bounding boxes. The model determines the grid in which the center of an object lies, and the bounding boxes within that grid are responsible for predicting the object. The loss function quantifies the discrepancy between the predicted bounding box coordinates, objectness, and class probabilities, and their ground truth values.

The loss function used in the YOLO-v4 model comprises three primary components: bounding box location loss $\mathcal{L}_{CIoU}$, confidence loss $\mathcal{L}_{conf}$, and classification loss $\mathcal{L}_{cla}$. The formula of the loss function $\mathcal{L}$ is expressed as follows:

$$\mathcal{L} = \boldsymbol{\lambda_1}\mathcal{L}_{conf} + \boldsymbol{\lambda_2}\mathcal{L}_{cla} + \boldsymbol{\lambda_3}\mathcal{L}_{CIoU} \tag{1}$$

where $\boldsymbol{\lambda}$ is the balance coefficient.

The confidence loss can be computed by utilizing the binary cross-entropy loss function, which can be expressed as:

Table 1: Hardware and software configurations

Component	Configuration
CPU	Intel(R) Xeon(R) Gold 5218
GPU	2× NVIDIA Quaro RTX 5000
RAM	64*Go*
Operating system	Windows 10
Development environments	Python 3.8, CUDA 11.4 and cuDNN 8.1

$$\mathcal{L}_{conf} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathcal{W}_{ij}^{obj}\left[\hat{C}_i^j log\left(C_i^j\right) + \left(1-\hat{C}_i^j\right) log\left(1-C_i^j\right)\right] - \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathcal{W}_{ij}^{obj}\left[\hat{C}_i^j log\left(C_i^j\right) + \left(1-\hat{C}_i^j\right) log\left(1-C_i^j\right)\right] \quad (2)$$

In Eq. 2 , S^2 represents the total number of grids in the image. B denotes the number of bounding boxes associated with each grid. $\mathcal{W}_{ij}^{obj}$ indicates whether an object exists in the grid cell. C_i^j refers to the confidence score of the *j-th* bounding box predicted by the *i-th* grid.

The classification loss is determined using the following formula:

$$\mathcal{L}_{cla} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathcal{W}_{ij}^{obj}\sum_{c=1}^{C}\left[\hat{p}_i^j(c) log\left(p_i^j(c)\right) + \left(1-\hat{p}_i^j(c)\right) log\left(1-p_i^j(c)\right)\right] \quad (3)$$

where $\hat{p}_i^j(c)$ (resp. $(p_i^j(c))$ represents the true (resp. predicted) probability that the predicted bounding box belongs to the class C.

To ensure faster convergence and improved regression accuracy during training, the YOLO-v4 model incorporates the Complete-Intersection over Union (CIoU) loss. In contrast to the standard IoU loss, the CIoU loss ($\mathcal{L}_{CIoU}$) accounts for geometric factors such as aspect ratio and the normalized distance between the centers of the ground truth and predicted bounding boxes. The CIoU loss is defined as follows:

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2\left(\boldsymbol{B_c^G}, \boldsymbol{B_c^P}\right)}{d^2} + \alpha\nu \quad (4)$$

where $\rho^2\left(\boldsymbol{B_c^G}, \boldsymbol{B_c^P}\right)$ is representing the Euclidean distance between the center points of ground truth predicted bounding boxes ($\boldsymbol{B_c^G}$ and $\boldsymbol{B_c^P}$, respectively). d represents the minimum diagonal distance of the two bounding boxes. $\nu = \frac{4}{\pi^2}\left(\arctan\frac{\boldsymbol{B_w^G}}{\boldsymbol{B_h^G}} - \arctan\frac{\boldsymbol{B_w^P}}{\boldsymbol{B_h^P}}\right)^2$ and $\alpha = \frac{\nu}{(1-IoU)+\nu}$ represent the trade-off factors used to assess the aspect ratio consistency, where, $\boldsymbol{B_w^P}$ (resp. $\boldsymbol{B_w^G}$) and $\boldsymbol{B_h^P}$ (resp. $\boldsymbol{B_h^G}$) are representing the width and height of the predicted (resp. the ground truth) bounding box.

To optimize performance, we created and trained two distinct models based on the YOLO-v4 architecture, with each model dedicated for one of the camera types. The models were named YOLO-CC (referring to the cooled camera) and YOLO-UC (referring to the uncooled camera). The hardware and software configurations used for models training and testing are presented in Tab. 1. The training was performed for 2500 iterations with a batch size of 12. The models are optimized using Adam with a learning rate of 0.0001.

To evaluate the models, precision, recall and F1-score are used and represented by Eqs. 5-7, respectively:

$$P = \frac{TP}{TP + FP} \quad (5)$$

$$R = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{7}$$

where, TP (True positive) represents the count of correctly detected axles, while FP (False positive) refers to other parts of the train (e.g., exhaust pipe) that are incorrectly detected as axles. On the other hand, FN (False negative) represents the number of axles that are missed or not detected.

2.3 Hot box tracking using Kalman filter

The Kalman filter is a typical tracking algorithm. It uses a recursive mathematical approach to estimate the state of a system being tracked over time based on incomplete and noisy measurements. This filter is widely employed in various applications. In the context of tracking hot wheels and hot axle bearings, the Kalman filter is employed in conjunction with the YOLO object detection method to accurately track their positions.

The Kalman filter operates by predicting the future position of the hot boxes based on their previous estimated state and the underlying dynamics of the system. It then corrects these predictions using the new measurements (the measurements, in this paper, represent the positions of the hot boxes detected by YOLO), continuously updating its estimation as new frames are processed. By iteratively refining the estimates, the Kalman filter compensates for the uncertainties and inconsistencies given by YOLO detections, resulting in a more accurate and robust tracking solution. Here, the centroids of the bounding boxes were used as the feature values to describe moving hot boxes.

The Kalman filter consists of two main steps: the prediction step and the update step. The prediction step involves two important sub-steps: state prediction and error covariance prediction. These sub-steps play a crucial role in estimating the current state, $\hat{x}_k^-$, of a system based on previous measurements and predictions. The state prediction equation, defined by Eq. 8, provides an estimation of the system's state, at a given time step, by incorporating knowledge of the system's dynamics. The error covariance prediction equation, defined by Eq. 9, estimates the uncertainty or covariance of the predicted state. It calculates the predicted error covariance matrix, P_k^-, using the previous error covariance matrix, P_{k-1}, F_k, and the process noise covariance matrix, Q_k, which represents the uncertainty or disturbance in the system dynamics that is not accounted for in the state transition model.

$$\hat{x}_k^- = F_k \hat{x}_{k-1} + \omega_{k-1} \tag{8}$$

where, F_k is the state transition matrix and ω_{k-1} is the noise process.

$$P_k^- = F_k P_{k-1} F_k^T + Q_k \tag{9}$$

In the update step, the predicted state and its associated uncertainty are refined based on available hot box positions detected by YOLO. It consists of several key equations. The residual equation, given in Eq. 10, calculates the difference between the detected hot box position value, z_k, and the predicted one based on the current state estimate. The residual covariance equation, given in Eq. 11, computes the uncertainty associated with this residual. The Kalman gain, denoted K_k and calculated with Eq. 12, determines the weight given to the detected hot box positions, in the update process. Next, the predicted state, $\hat{x}_k^-$, is adjusted using Eq. 13 based on the Kalman gain and the residual. In the same manner, the error covariance is updated, by Eq. 14, using the optimal, K_k, reflecting the incorporation of the measurement information into the state estimation.

$$y_k = z_k - H_k \hat{x}_k^- \tag{10}$$

$$S_k = H P_k^- H_k^T + R_k \tag{11}$$

$$K_k = P_k^- H_k^T S_k^{-1} \tag{12}$$

$$\hat{x}_k = \hat{x}_k^- + K_k y_k \tag{13}$$

$$P_k = (I - K_k H_k) P_k^- \tag{14}$$

Where, H_k is the measurement matrix, R_k is the measurement noise covariance matrix.

2.3.1 Feature matching for individual hot box tracking

Assigning tracker ID for a detected hot box and maintaining it across consecutive frames is crucial for robust and reliable tracking. In our paper, this is achieved by the following procedure:

- Since the trains travel in straight-line trajectory, we decided to define an *ID allocation area* on the frames. This area serves to create a new potential tracker for each detected hot box whose center falls within the area boundaries. However, false positives can also occur within this designated area, so additional measures were implemented. We computed a similarity score, using the structural similarity index (SSIM),[20] between the presumed hot box, detected within the area in the current frame, with those detected in the two next frames to provide more insurance. If the scores are below a predefined threshold, the potential tracker is assigned by false detection and is eliminated.
- To maintain the trackers IDs, and to update their state with the adequate YOLO detections, we employed a method that involved calculating the absolute difference between the centroid position of each detected hot box and the predicted positions by the existing trackers. Subsequently, each detection is assigned to the tracker that exhibits the closest proximity.
- In cases where YOLO fails to detect hot boxes, i.e. the tracker is not assigned to any detection, the state of the tracker is updated by the Kalman filter and its ID is retained.
- If the number of times a tracker remains unassigned by a detection exceeds the predefined threshold, it is considered to have disappeared and is deleted.

3. RESULTS AND DISCUSSION

In this section, we present a comprehensive analysis of the hot box detection results obtained from YOLO-CC and YOLO-UC. These models were tested on five IR image sequences capturing the passage of five different trains, including four freight trains and one passenger train. These sequences, taken from both uncooled and cooled cameras, are selected to include both daytime and nighttime experiments, featuring scenarios where the sun reflects off the train body or is pointed directly towards the camera.

Visual representations of the detection results, are partially illustrated in Fig. 2 and 3, respectively, with the Kalman filter tracking and counting results highlighted in orange color. It can be observed, in some frames depicted in the figures, a situation where the sensor of the cameras is overwhelmed by the strong infrared radiation emitted by the sun, resulting in significant loss of details. Additionally, considering that many trains, particularly freight trains, rely on diesel combustion engines for propulsion, the heat generated by these engines are intensified and channeled through the exhaust pipe, resulting in a pronounced thermal signature in the IR images. However, despite these conditions, both YOLO-CC and YOLO-UC have shown great capability to detect and distinguish only the hot boxes, while disregarding these specific parts.

Table. 2 provides an evaluation of the YOLO-CC and YOLO-UC performances. The table includes the number of frames used for testing the models, the evaluation metrics results and the inference speed. It is obvious that all metrics exhibit high performances greater than 99.6%. The F1-scores, for both the cooled and uncooled cameras, exceed 99.7% indicating the great balance that the models offer between precision and recall.

Finlay, it is worth-noting that, the integration of Kalman filter tracking into the YOLO has resulted in a notable improvement of the F1-score, approaching 100%. This is because the Kalman filter predictions contributed to the validation or rejection of the hot box detection results, thereby enhancing the overall performance.

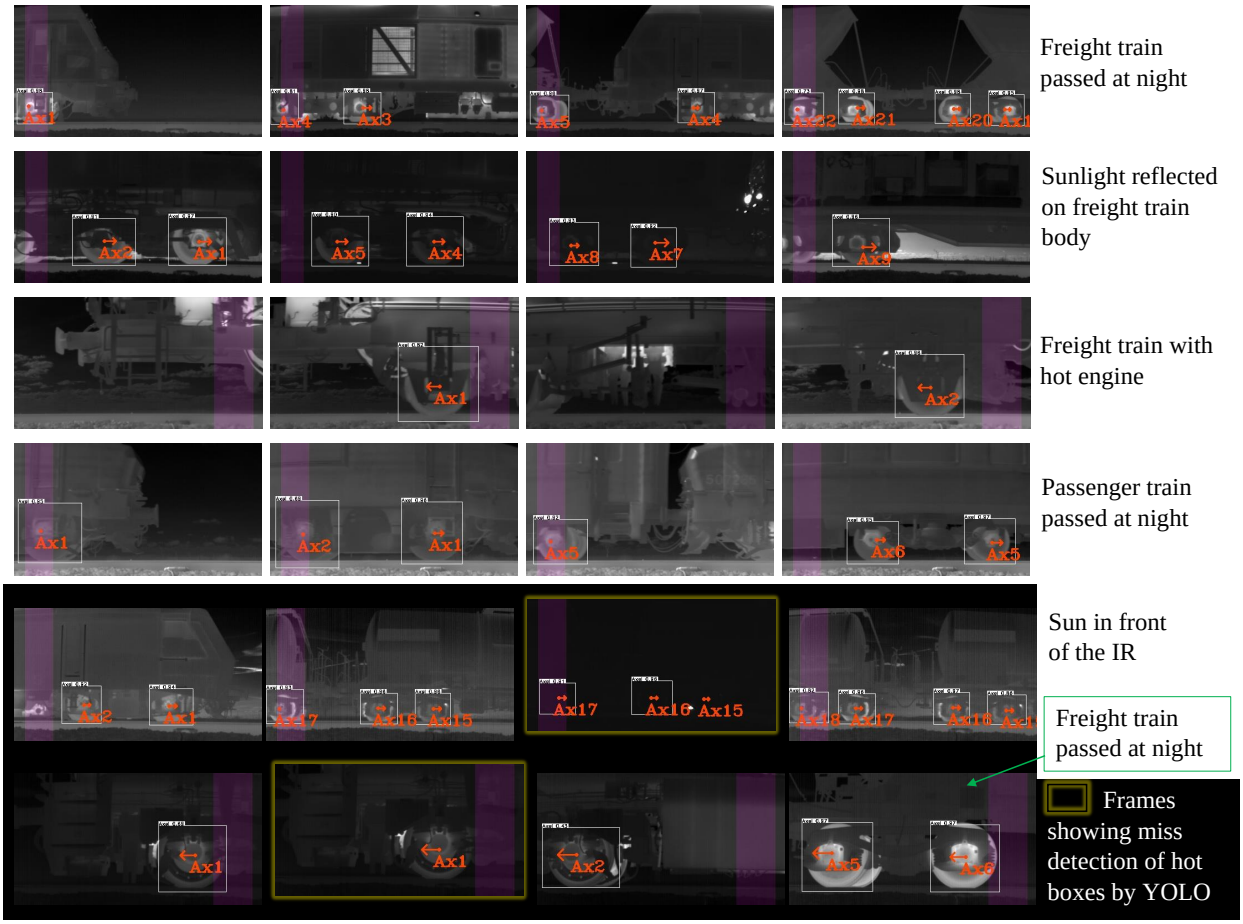

Figure 2: Hot box detection results on selected frames taken from cooled camera using YOLO-CC

Table 2: YOLO performance on IR thermal image sequences taken by cooled and uncooled cameras, which are partially shown in Fig. 3 and 2; conf-thresh = 0.40; IOU = 0.38

Results (%) / Metrics	IR camera	
	Cooled	Uncooled
	Number of tested frames 13910	Number of tested frames 6630
Precision	99.93	99.88
Recall	99.72	99.64
F1-score	99.82	99.76
Inference speed	63 fps	

4. CONCLUSION

In the railroad industry, timely detection of overheated rail-car wheels and bearings is of utmost importance for ensuring safety. Motivated by this concern, we have introduced an infrared-imaging-based computer vision method that combines the YOLO object detector with the Kalman filter as a tracker. This integration addresses the challenges associated with providing a cost-effective hot box detector that is easy to maintain and can operate autonomously without relying on external facilities. By leveraging the YOLO object detector, we have achieved efficient hot box detection even under difficult conditions. The Kalman filter complements the detection process by providing robust tracking capabilities, enabling continuous monitoring of the detected hot boxes across consecutive frames. This fusion enhances the overall accuracy and reliability of the proposed hot box detection and counting method. In future work, we will focus on two primary objectives. Firstly, we aim to enhance the

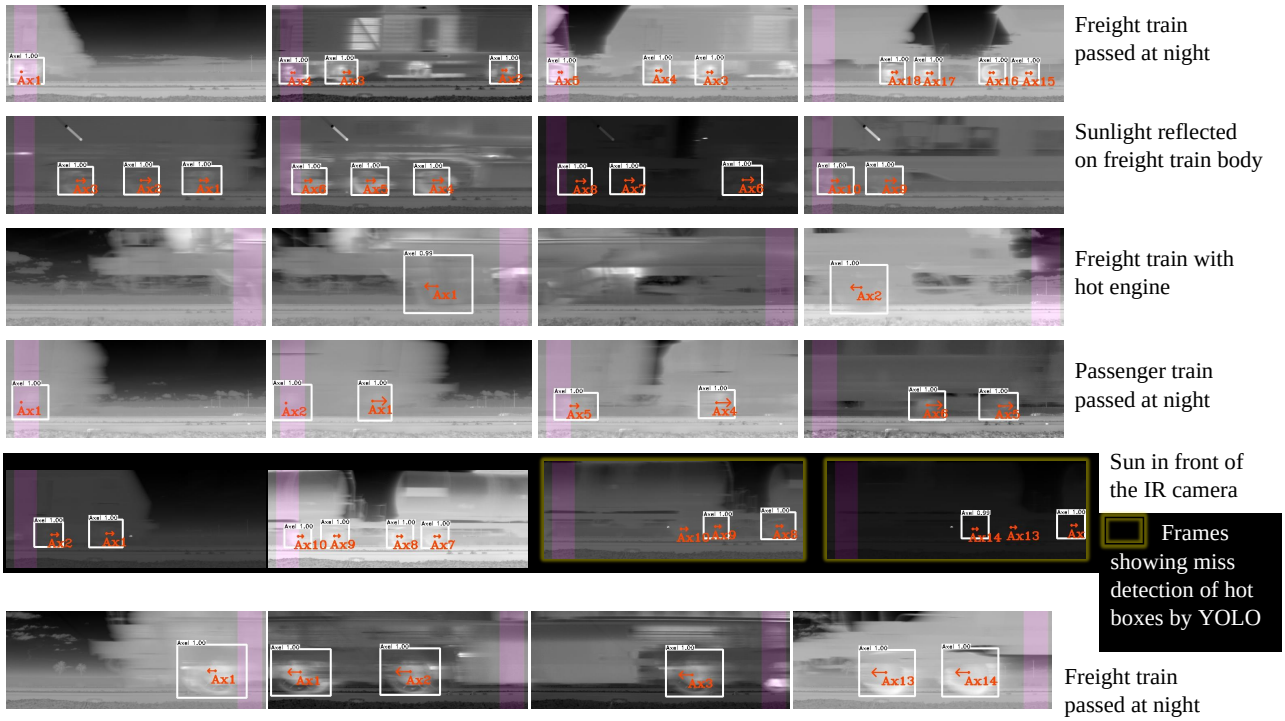

Figure 3: Hot box detection results on selected frames taken from uncooled camera using YOLO-UC

blur effect in the uncooled-based IR images. Secondly, we plan to propose a method for accurately estimating the temperature of the hot boxes from the IR images and perform a comparative analysis with the current temperature measurement system.

ACKNOWLEDGMENTS

Authors would like to acknowledge the BRIGHTER project and SNCF Reseau for supporting this work. BRIGHTER has received funding from the KDT Joint Undertaking (JU) under grant agreement No 101096985. The JU receives support from the European Union's Horizon Europe research and innovation program and France, Belgium, Portugal, Spain, Turkey.

REFERENCES

[1] Falamarzi, A., Moridpour, S., and Nazem, M., "A review on existing sensors and devices for inspecting railway infrastructure," *Jurnal Kejuruteraan* **31**(1), 1–10 (2019).

[2] Tarawneh, C., Aranda, J., Hernandez, V., Crown, S., and Montalvo, J., "An investigation into wayside hot-box detector efficacy and optimization," *International Journal of Rail Transportation* **8**(3), 264–284 (2020).

[3] Deilamsalehy, H., Havens, T. C., Lautala, P., Medici, E., and Davis, J., "An automatic method for detecting sliding railway wheels and hot bearings using thermal imagery," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* **231**(6), 690–700 (2017).

[4] Toullier, T., Dumoulin, J., and Bourgeois, V., "Comparative study of moving train hot boxes pre-detection and axles counting by in-situ implementation of two infrared cameras," in [*QIRT Asia 2019-Quantitative InfraRed Thermography Conference*], (2019).

[5] Ahuja, N. and Barkan, C. P., [*Machine vision for railroad equipment undercarriage inspection using multi-spectral imaging*], IDEA Programs, Transportation Research Board (2007).

[6] Kim, M.-S., Oh, S.-C., Kim, G.-Y., and Kwon, S.-J., "Underbody component monitoring system of railway vehicles using the infra-red thermal images," in [*2014 International SoC Design Conference (ISOCC)*], 222–223, IEEE (2014).

[7] Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence* **38**(1), 142–158 (2015).

[8] Girshick, R., "Fast r-cnn," in [*Proceedings of the IEEE international conference on computer vision*], 1440–1448 (2015).

[9] Girshick, R., "Fast r-cnn," in [*Proceedings of the IEEE international conference on computer vision*], 1440–1448 (2015).

[10] Shi, J., Chang, Y., Xu, C., Khan, F., Chen, G., and Li, C., "Real-time leak detection using an infrared camera and faster r-cnn technique," *Computers & Chemical Engineering* **135**, 106780 (2020).

[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "Ssd: Single shot multibox detector," in [*European conference on computer vision*], 21–37, Springer (2016).

[12] Ding, L., Xu, X., Cao, Y., Zhai, G., Yang, F., and Qian, L., "Detection and tracking of infrared small target by jointly using ssd and pipeline filter," *Digital Signal Processing* **110**, 102949 (2021).

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

[14] Redmon, J. and Farhadi, A., "Yolo9000: better, faster, stronger," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 7263–7271 (2017).

[15] Redmon, J. and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767* (2018).

[16] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934* (2020).

[17] Zhang, Z., He, T., Zhang, H., Zhang, Z., Xie, J., and Li, M., "Bag of freebies for training object detection neural networks," *arXiv preprint arXiv:1902.04103* (2019).

[18] Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., and Yeh, I.-H., "Cspnet: A new backbone that can enhance learning capability of cnn," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*], 390–391 (2020).

[19] Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J., "Path aggregation network for instance segmentation," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 8759–8768 (2018).

[20] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing* **13**(4), 600–612 (2004).