IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 23 February 2024, accepted 12 March 2024, date of publication 18 March 2024, date of current version 25 March 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3378522

RESEARCH ARTICLE

A Novel Vision-Based Truck-Lifting Accident Detection Method for Truck-Lifting Prevention System in Container Terminal

ZHIYU JI[1,*], KAIKAI ZHAO[2,3,*], ZHAOXIANG LIU[2,3,*], HUAN HU[2,3], ZHONGHENG SUN[4], AND SHIGUO LIAN[2,3], (Member, IEEE)
[1]Nanjing Port Longtan Container Ltd., Nanjing Port (Group) Ltd., Nanjing 210011, China
[2]Unicom Digital Technology, China Unicom, Beijing 100013, China
[3]AI Innovation Center, China Unicom, Beijing 100013, China
[4]China United Network Communications Ltd., Nanjing Branch, Nanjing 210008, China

Corresponding authors: Zhaoxiang Liu (liuzx178@chinaunicom.cn), Zhongheng Sun (js-sunzh@chinaunicom.cn), and Shiguo Lian (liansg@chinaunicom.cn)

This work was supported by Beijing Association of Science and Technology Outstanding Engineer Growth Plan.
*Zhiyu Ji, Kaikai Zhao, and Zhaoxiang Liu contributed equally to this work.

ABSTRACT Truck-lifting accidents are common in container-lifting operations. The most common truck-lifting prevention system is based on lidar scanners which are very expensive, and it is more promising to use cameras to detect truck-lifting accidents than to use lidar scanners. However, little work has been conducted on visual detection of truck-lifting accidents, and the only previous visual approach fails in the common scenario where wheel hubs in images appear as non-standard or incomplete circles. To this end, this paper proposes a novel vision-based truck-lifting accident detection method for truck-lifting prevention system, which is free from the disturbance of distorted or incomplete wheel hubs in the image. The main idea of the proposed method is to utilize a deep learning-based object detection model to detect the truck body within which to extract many key-points whose vertical displacements are tracked to determine whether the truck is lifted. Based on this idea, the workflow for truck-lifting accident detection is delicately constructed. In addition, a YOLOv5-based modified detection model is proposed to reduce the computation cost of container and truck body detection, achieving 38.5% increase in inference speed on a single industrial personal computer without performance decrease. The experimental results demonstrate that the proposed truck-lifting accident detection method is capable of accurately recognizing the truck-lifting operations with the recall rate of 100% and the false alarm rate of 0.42%.

INDEX TERMS Truck-lifting prevention system, truck-lifting accident detection, deep learning, object detection, key-points.

I. INTRODUCTION

Container terminals are important infrastructures for container transportation in global trade. In recent years, automated container terminals become more and more popular in reducing labor costs, improving operation efficiency, and promoting worker safety [1], [2], [3], [4], [5]. In the terminal operation process, containers need to be transferred between various storage areas. The trucks carry the containers from one place to another, and the container-lifting equipment such as the gantry cranes lift the containers from the trucks and place them in the container yard. During the container-lifting operation, there is one common accident called truck-lifting accident that occurs when the container lock pins are not released properly (as shown in Fig. 1). The truck would be lifted with the container together, and this would cause damage to the container, the truck, and the lifting ropes and endanger the safety of on-site workers. Therefore,

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

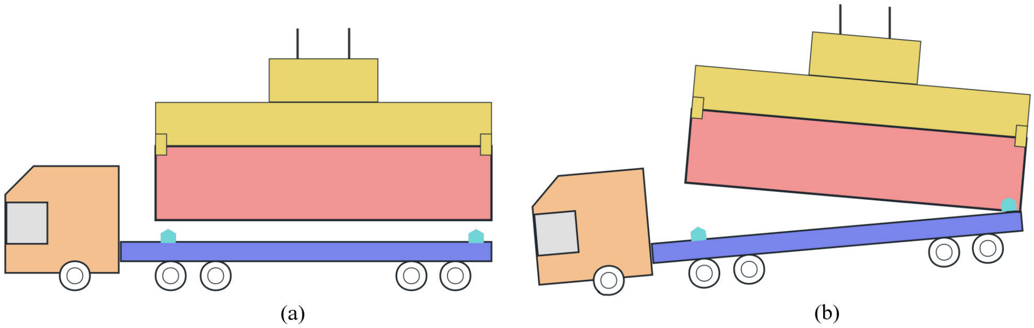


FIGURE 1. The diagram of container-lifting operation. (a) Normal lifting. (b) Truck-lifting accident.

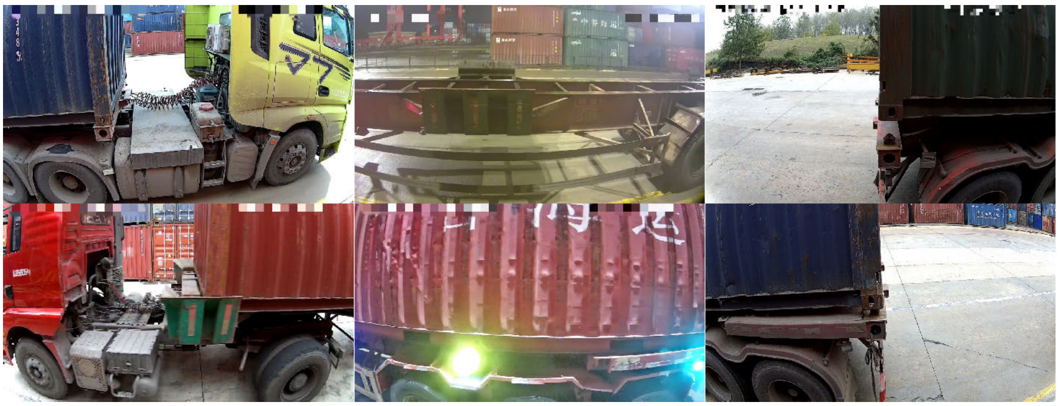


FIGURE 2. Representative images collected from real application scenarios, containing images of the head region of the truck body (first column), images of the middle region of the truck body (second column), and images of the tail region of the truck body (third column).

an automated accident detection method is required to prevent the accident.

The most common truck-lifting prevention system is based on lidar scanners [6], [7], [8]. The lidar scanner scans the contour of the truck and the container and builds their corresponding 3D models. By analyzing the relative positional relationship of the truck and the container, the truck-lifting accident could be detected. However, this kind of system requires a high-precision lidar scanner which is very expensive, which significantly hinders its widespread adoption and implementation.

Compared to a lidar scanner, a camera is a quite low-cost sensor. Moreover, a camera could provide higher resolution information than a lidar scanner, and achieve higher measurement accuracy. When employed with deep convolutional neural networks, it has the ability to recognize complex visual features. In addition to the above advantages, a camera is capable of supporting real-time remote monitoring or other visual applications (such as container code recognition, truck scheduling, and container damage identification). In case of accident detection algorithm failure, the system could switch to manual monitoring. This is a very useful feature that the lidar scanner does not have.

In a word, it is more promising to use cameras to detect truck-lifting accidents than to use lidar scanners. However, very little work has been done on visual truck-lifting accident detection. The article [9] firstly proposes a vision-based method to detect truck-lifting accidents. It consists of an object detector to detect the wheel hubs, a Hough circle detector [10] to refine the wheel hubs detection, and a bounding box tracking module to monitor the movement of the wheel hubs to determine whether the truck is lifted. However, it tends to fail when Hough circle detector cannot detect the wheel hubs and this case often happens [9]. Several images collected from real application scenarios are presented in Fig. 2, clearly demonstrating instances where the Hough circle detector encounters failure due to the distorted, incomplete, or even unseen wheel hubs. Distorted wheel hubs refer to wheel hubs that do not appear as standard circular shapes in the images. So there is still a large room to explore in visual truck-lifting accident detection. In addition, visual truck-lifting accident detection under limited computing resources also needs to be explored, because lightweight and low-cost computing devices are more desirable in real applications.

To this end, we propose a novel vision-based method for truck-lifting accident detection. The main idea of our method is to utilize a visual object detector to detect the truck body, including the truck chassis and the tire if the tire exists, within which to extract some key-points whose vertical displacements are tracked to determine whether the truck is lifted.

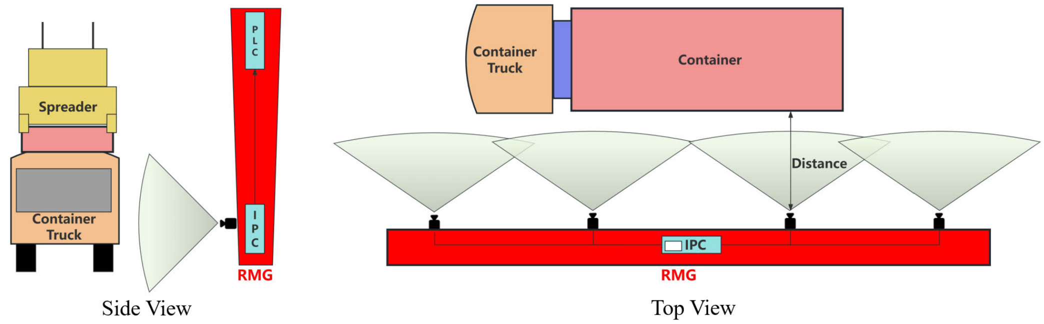


FIGURE 3. Side and top views of the hardware of our system.

FIGURE 4. The real hardware of our system. Cameras are in the green rectangle.

The contributions of this paper can be summarized as follows:

(1)We propose a novel vision-based method for truck-lifting accident detection. The main idea of our method is to utilize a visual object detection model to detect the truck body within which to extract some key-points whose vertical displacements are tracked to determine whether the truck is lifted. Based on the idea, the workflow for truck-lifting accident detection is constructed, and the experimental results demonstrate the generalization and robustness of our method.

(2)Aiming to reduce the computation cost of container and truck body detection, we propose a YOLOv5 [11] based modified detection model which merely has the top detection head with only one anchor while the other two detection heads are removed. Compared with the original YOLOv5 model, it achieves 77 fps in inference with the input resolution of 608×608 on a single industrial personal computer, gaining 38.5% speed acceleration without performance decrease.

II. RELATED WORK

With the rapid development of computer vision and deep learning, vision-based detection and measurement technologies are widely used in industry [12], [13], [14], including container terminals [5], such as container crane spreader measurement [15], container code recognition [16], [17] and container corner casting recognition [18], [19].

Utilizing cameras instead of lidar scanners to detect truck-lifting accidents is more promising, however, this study is still in its early stage and very little work has been done. The article [9] proposes to employ a single shot multi-box detector (SSD) [20] to detect the wheel hubs of the truck, then utilize a Hough circle detector to improve the detection of the wheel hubs, finally adopt a Deep Sort [21] based tracking module to measure the displacement of the bounding box of the wheel hubs to determine whether the truck is lifted. However, the key module of this method, i.e., Hough circle detector, often suffers from defaced tire, incomplete tire, or image distortion, and such factors lead to poor robustness in truck-lifting accident detection. Considering the truck body has a larger span than wheel hubs and always appears in the view of cameras, obviously, detecting the truck body instead of wheel hubs is more robust. Besides, tracking some significant key-points within the truck body is more robust than tracking the bounding box of the wheel hubs, because the truck detector is able to stably output a body rectangle box within which some key-points can be constantly extracted, no matter whether the tire is distorted, incomplete or even unseen. In a word, compared to the method in [9], our method could be free from the disturbance of distorted, incomplete, or unseen wheel hubs in the image.

There are also two kinds of intuitive approaches to detecting truck-lifting accidents. The first approach is object detector-only. However, it has an inherent limitation in accurate boundary localization, i.e., bounding box jitter, that could potentially affect the truck-lifting accident detection accuracy, as mentioned in [9]. The other approach relies solely on key-points tracking technique which is widely used in many fields, such as the internet of things, robotics, and autonomous driving. However, it is challenging to identify whether the truck is present, where the truck body is, and whether the truck is lifted because there are a lot of interfering objects such as on-site workers, cars, forklifts, and other types of equipment in the real operating environment.

III. METHOD

Object detectors provide category and location information that guide point-level trackers, allowing them to concentrate on interesting regions, thereby reducing computational costs and enhancing precision. In turn, point-level trackers compensate for the deficiency of detectors in accurate object boundary localization. Consequently, integrating an object detector with a key-point tracker presents a theoretically sound approach for detecting truck-lifting accidents.

The main idea of our method for truck-lifting detection is to utilize a visual object detection model to detect the truck body within which to extract some key-points whose vertical displacements are tracked to determine whether the truck is lifted. To achieve this idea, selecting a proper image as reference frame, obtaining a stable truck region from the reference frame, extracting some key-points within the truck region, establishing the reference information as the starting point of tracking, and identifying truck-lifting operation are essential for our method. So the workflow of our system needs to be designed delicately. In this section, the hardware setup for our truck-lifting prevention system will be introduced first, then the workflow of our method and the method for truck and container detection will be introduced in detail.

A. HARDWARE SETUP

Fig. 3 presents the schematic diagram of our truck-lifting prevention system, and Fig.4 presents the real system hardware. Considering that the container mainly has two different specifications, i.e., 20 feet and 40 feet, four internet protocol cameras with resolution 1280×720 are installed at the leg of the rail-mounted container gantry crane(RMG) to capture the images of truck and container. The captured images are all processed on a single industrial personal computer (IPC) with an Intel 1185G7 CPU (4 Cores, 8 threads) built-in an integrated graphics card (Intel Iris Xe Graphics), to determine whether the container truck is lifted or not. Once the truck-lifting accident is detected by any one of the four cameras, the IPC will immediately send a signal to the RMG controller, i.e., the programmable logic controller (PLC), to stop lifting the container spreader.

B. WORKFLOW

For each camera, its video stream is processed through the same workflow independently as shown in Fig. 5.

Initially, the system decodes the video stream and samples frames at a constant rate of 15 frames per second to obtain the image sequence, denoted as $F = [F_1, F_2, \ldots]$. Subsequently, the router dispatches F into the corresponding processing branch (“free branch” or “busy branch”) based on the current system state (“free state” or “busy state”). When the truck is not ready for the gantry cranes to lift the container, the system is in “free state”, otherwise in “busy state”. In reference information construction step or completion identification step, the system state may be updated.

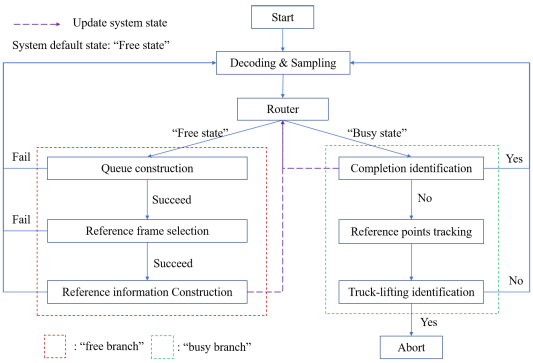


FIGURE 5. Wokflow of system method.

1) “FREE BRANCH”

The “free branch” (as shown in the red box of Fig. 5) is responsible for monitoring whether the truck has stopped at the right location and is ready for lifting operation and constructing the reference information with regard to which used to track the movement of the truck. It consists of three steps: the construction of queue, the selection of reference frame, and the construction of reference information.

a: QUEUE CONSTRUCTION

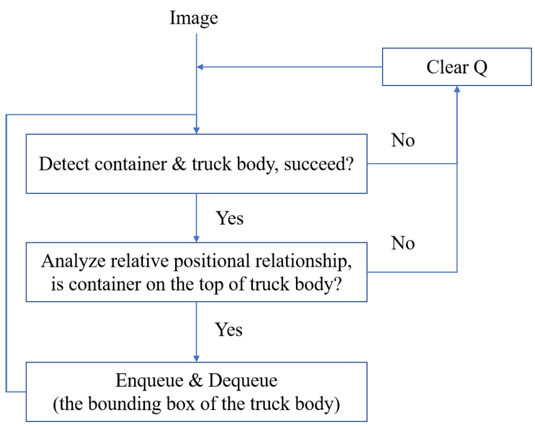


FIGURE 6. Flow chart of queue construction.

To monitor whether the truck has stopped at the right location, we construct a queue with a fixed length of q, which consists of the locations of the truck at each moment. The queue is defined as $Q = [b_1, b_2, \ldots, b_i, \ldots, b_q]$. b_i represents the bounding box of the truck body extracted from its corresponding frame F_i, b_1 is the head of the Q, and b_q is the tail of the Q.

Fig. 6 shows how the queue is constructed. The Q must meet three requirements: 1) its corresponding frames must be continuous; 2) each corresponding frame must contain both the container and the truck body; 3) the container of each corresponding frame must be on the top of the truck body. To ensure the last two requirements, a modified YOLOv5 detector (YOLOv5s-light, which will be detailed in section III-C) is utilized to detect the container and the truck body, whose bounding boxes are used to analyze the relative positional relationship between them.

Once the Q is built successfully, it means that the truck loading the container has arrived near the lifting location, and the next step is to monitor whether the truck has stopped at the right location and select an appropriate frame with regard to which used to track the movement of the truck. We call this appropriate frame reference frame, represented by the F_r.

b: REFERENCE FRAME SELECTION

The reference frame can be selected based on the intersection over union (IoU) between b_1 and b_q. When $IoU(b_1, b_q) > 0.95$, we infer that the truck has been parked in the right location, the reference frame is established as $F_r = F_q$, and the truck body bounding box in the reference frame is established as $b_r = b_q$.

Once the reference frame is selected successfully, that signifies the truck has stopped at the right location waiting for lifting container and the next step is to construct the reference information with regard to which used to track the movement of the truck.

c: REFERENCE INFORMATION CONSTRUCTION

The reference information comprises the truck body bounding box, i.e., b_r, key-points coordinates, and key-points descriptors. These key-points are extracted within the truck body region in F_r using ORB [22].

After the reference information is established, the system will switch from "free state" into "busy state", and the router will dispatch the next frame into "busy branch".

2) "BUSY BRANCH"

The "busy branch" (as shown in the green box of Fig. 5) is responsible for monitoring whether the lifting operation is complete and whether the truck is lifted based on the reference information. It comprises three steps: the identification of completion, the tracking of reference points, and the identification of truck-lifting.

a: COMPLETION IDENTIFICATION

In this step, the truck body is detected using YOLOv5s-light in the current frame, and the output bounding box is denoted as b_c. Then, for a consecutive sequence of 10 frames, if $IoU(b_c, b_r) < 0.5$ is met, we infer that the lifting operation is complete and the truck is going to leave. Then, the system switches into "free state". Otherwise, the system goes to the reference points tracking step.

b: REFERENCE POINTS TRACKING

Reference points refer to the key-points in the reference information in section III-B1.c, and they are denoted as a set $P_r = \left[\left(x_1^r, y_1^r\right), \left(x_2^r, y_2^r\right), \ldots, \left(x_i^r, y_i^r\right), \ldots, \left(x_p^r, y_p^r\right)\right]$, where $\left(x_i^r, y_i^r\right)$ is the coordinate of the i_{th} reference point, and p presents the number of reference points. Reference points tracking is the process of tracking these points within each subsequent frame.

To update the reference points tracking results in the current frame, we define a temporary buffer $P_c = \left[\left(x_1^c, y_1^c\right), \left(x_2^c, y_2^c\right), \ldots, \left(x_i^c, y_i^c\right), \ldots, \left(x_p^c, y_p^c\right)\right]$. At the beginning of tracking, each $\left(x_i^c, y_i^c\right)$ in P_c is initialized with the coordinate of the corresponding reference point, i.e., $x_i^c = x_i^r, y_i^c = y_i^r$. In each subsequent frame, we first extract some key-points within the truck body bounding box using ORB in the current frame, then we utilize a feature matching-based method, i.e., brute-force matcher in OpenCV [23], to track the reference points. Finally, for those successfully matched reference points, the corresponding coordinates in the buffer P_c will be updated to be the coordinates of the matched points in the current frame.

c: TRUCK-LIFTING IDENTIFICATION

After reference points tracking, we analyze the movement of all reference points in the current frame to identify the truck-lifting result as either "danger" (there is the truck-lifting risk) or "safe" (there is no truck-lifting risk). Because the variation of the distance (as shown in Fig. 3) between the truck and the camera is very small, the mapping between the vertical displacement in pixels and the displacement in centimeters is constant. Therefore, we can use the vertical displacement in pixels to determine whether the truck is lifted.

Firstly, the vertical displacement d_i of the i_{th} reference point is computed by (1).

$$d_i = y_i^c - y_i^r \tag{1}$$

Then, displacements of all reference points can be obtained and denoted by $D = \left[d_1, d_2, \ldots, d_i, \ldots, d_p\right]$. Following that, the movement of the i_{th} reference point is computed by (2).

$$m_i = \begin{cases} 1, & \text{if } d_i > T_d \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where T_d is the distance threshold, controlling our system's sensitivity. Ultimately, we define M as the sum of all m_i, i.e., $M = \sum_{i=1}^{p} m_i$, and the truck-lifting result is computed by (3).

$$r = \begin{cases} 1, & \text{if } M > \max\left(T_n^a, T_n^r p\right) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $r = 0$ and $r = 1$ indicate that the truck-lifting result of the current frame is "safe" and "danger", respectively. T_n^a is the absolute quantity threshold, and T_n^r is the relative quantity threshold. If $r = 1$, our system sends a signal to PLC to stop the lifting operation.

C. DETECTION MODEL

Our method for truck-lifting accident detection is based on a visual object detector. The object detection model detects the container and the truck body for each frame in the queue construction step or the completion identification step, and this would cost a lot of computing resources.

The state-of-the-art (SOTA) in the object detection field includes two-stage detectors such as Faster R-CNN series [24] and one-stage detectors like SSD series and YOLO series. The two-stage detectors have high detection accuracy but with low inference speed and high computational demand. The one-stage detectors achieve decent detection accuracy with faster inference speed. Compared with SSD series, YOLO series has a better balance in accuracy and speed, which is widely used in industry and academia. After thorough comparison (detailed in section IV-A3), the YOLOv5s is adopted in our method. Considering that the images of four cameras are all processed on a single IPC with limited computing resources, this would be still a big burden for the IPC, so the detection model needs to be further improved to reduce the computation cost for the IPC.

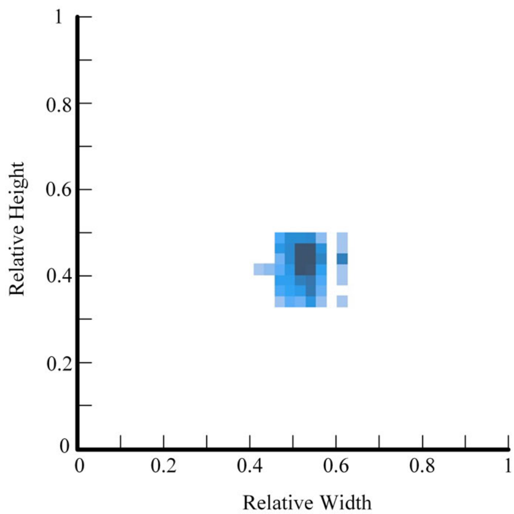


FIGURE 7. **The size and shape distribution of objects in our dataset.**

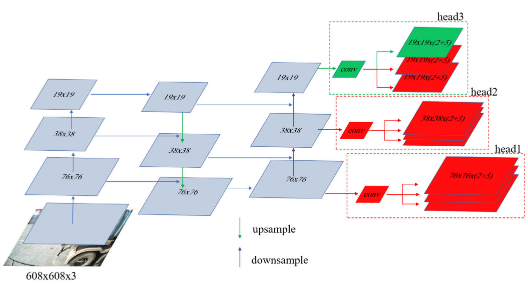


FIGURE 8. **The architecture of YOLOv5s-light. The red parts of the original YOLOv5s are pruned, and only one anchor is retained on the top feature map in our detector.**

Original YOLOv5s imposes nine anchors with different sizes and aspect ratios on three different scale feature maps to detect different objects. However, for our system, the distance between the truck and the camera varies little, and the lifting position relative to the camera is also fixed. This results in the size and the aspect ratio of the truck body and container in the image varies little. Besides, the truck body and the container have similar size and aspect ratios in the image, as shown in the first column and third column of Fig. 2. The size and shape distribution of all objects in our dataset is shown in Fig. 7. So there is no need to use nine different anchors on three scale feature maps to detect the container and the truck body.

The architecture of modified YOLOv5 (YOLOv5s-light) is shown in Fig. 8. Considering that the container and the truck body in image are relatively large, we only retain the top detection head while the bottom and middle detection head are removed, and only one anchor is imposed on the top feature map. The size of the anchor is set to be the average size of the container and the truck body in our training set. Theoretically, our proposed modification reduces the computational cost of the detection head by 98% in comparison with the original YOLOv5s model.

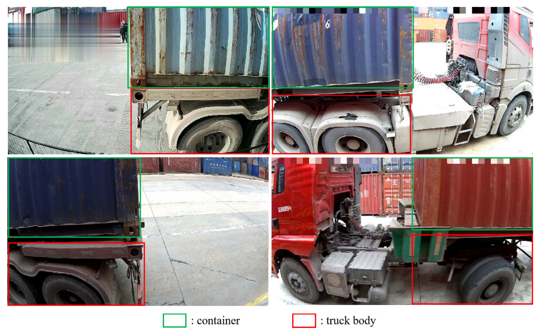


FIGURE 9. **Detection results of truck body and container.**

IV. EXPERIMENT

In this section, we will evaluate the modified object detection model first, and then evaluate the whole system.

A. DETECTION MODEL EVALUATION

1) TRAINING DATA

Using the hardware setup introduced earlier, we collected a total of 5000 images as training data(60%), validation data(20%), and test data(20%). Each image contains a part of the container and the truck body. All data are annotated by professional engineers. They are used for the training and testing of our container and truck body detector. It should be noted that images such as those in the second column of Fig. 2, which contain the middle region of the truck body, are not annotated.

2) TRAINING DETAILS

Our detection networks are trained with Pytorch [25]. The detection model is trained for 150 epochs from the YOLOv5s model pre-trained on coco datasets using SGD optimizer [26], with a learning rate of 0.01 and batch size of 128 on 8 NVIDIA V100 GPUs. The size of the input image is resized from 1280×720 to 608×608 using the adaptive resizing method, which maintains the original aspect ratio of objects. We suggest using at least 3000 images to train the truck detection model.

TABLE 1. Performance comparison including mAP, recall, and detection speed(in FPS) between different state-of-the-art models.

Method	mAP @0.5	Recall @0.5	mAP @0.5-0.95	mAP @0.95	FPS
Faster-RCNN	89.2	90.1	83.7	75.3	16
SSD	82.3	83.0	77.4	68.2	48
YOLOv5n	86.4	90.2	82.3	70.1	98
YOLOv5s	99.5	99.9	96.4	86.6	56
YOLOv5m	99.6	99.9	97.2	89.2	25
YOLOv5s-light (ours)	99.5	99.9	93.4	81.7	77

3) EXPERIMENTAL RESULTS

We compare our modified detection model with SOTA detectors, including Faster-RCNN, SSD, YOLOv5n, YOLOv5s, YOLOv5m in terms of mAP, recall, and inference speed. Fig. 9 shows the detection results of our model, and Tab. 1 shows the comparison result.

Because one single IPC needs to process data from four cameras at 60 frames per second, the system must achieve a minimum processing frame rate of 60fps. Taking into account the additional time consumed by key-points extraction and tracking, it is recommended for the detector to run at a speed over 70fps.

Among these detectors, YOLOv5n demonstrates a remarkable processing speed of 98fps, but this comes at the cost of a 13.1 mAP@0.5 decrease in detection precision when compared to YOLOv5s. On the other hand, YOLOv5m achieves the best detection accuracy, but its inference speed is much slower than our requirement. YOLOv5s presents an excellent trade-off between accuracy and speed, however, its inference speed falls a little below the desired threshold of 70fps, thereby motivating us to improve YOLOv5s for faster inference speed.

We can see that our detection model could run at the speed of 77 fps and achieve 38.5% acceleration in inference speed without performance decrease in mAP@0.5 and recall@0.5 compared with the original YOLOv5s model. This indicates that it is unnecessary to detect the container and truck body using multi-scale feature maps with nine anchors in our system. This is very significant for the scenario under limited computing resources. So our modified detection model can help to reduce the cost of the whole system. Notably, the decline in mAP@0.95 has a negligible impact on the truck-lifting accident detection performance, given that pixel-level key-points extraction and tracking are conducted in subsequent stages.

B. SYSTEM METHOD EVALUATION

1) TRAINING DATA

Relatively, truck-lifting accidents happen not very often in daily lifting operations, so it is very difficult to collect sufficient accident samples for our experiment and we collect truck-lifting accident samples by manually simulating the accidents. Considering all the cases that the lock pin is not released, there are fifteen combinations in total for the four lock pins. For each combination, constrained by the experimental loss of truck, container, and lifting rope, the truck-lifting operation is manually repeated only ten times, then we could get 150 videos for the truck-lifting operation in total. Besides, the videos for the normal container-lifting operation can be easily collected from daily container-lifting operations and 1000 lifting operation videos are collected in our experiment. All the videos are split into the training set, the validation set, and the test set. Therein, 100 truck-lifting videos and 650 normal operation videos are used as training (80%) and validation (20%) data, and the rest videos are used as test data.

2) TRAINING DETAILS

We utilize the training and validation videos to optimize the truck-lifting identification parameters introduced in section III-B in an iterative way within a predefined range. The optimal parameter combination can be obtained when the recall and false alarm rate reach the best, and here we list the suggested parameters: $q = 30$, $T_d = 20$, $T_n^r = 0.4$, and $T_n^a = 40$.

3) EXPERIMENTAL RESULTS

We evaluate our method for truck-lifting accident detection and our method achieves 100% recall (all 50 truck-lifting videos are recognized as "danger" successfully) and 0% false alarm (all 350 normal lifting videos are recognized as "safe" successfully).

Fig. 10 compares the reference points tracking process of a normal container-lifting operation and a truck-lifting accident operation. During the normal lifting operation (the first row), the truck body experiences a slight rebound at the early stage and then stays static. So we can see some short trajectories in the second column of Fig. 10(a). During the truck-lifting accident operation (the second row), the truck body continues to rise along with the container. So we can see that a large portion of reference points have a large vertical displacement in the second column and the third column of Fig. 10(b). We can see that our method still successfully detects the truck-lifting accident when wheel hubs are distorted, incomplete, or even unseen, while this is a very common failure in article [9].

4) DISCUSSION

In addition to the videos in the aforementioned dataset, our system has consistently operated in multiple container terminals and collected some additional test samples. In nearly 100000 normal lifting operations, our system mistakenly reported "danger" 42 times. In all 6 truck-lifting accidents, our system accurately identified the accidents and triggered the PLC to stop the lifting operation, thereby significantly avoiding potential accident losses. So, the recall and false alarm rates are updated to 100% and about 0.42%.

We analyzed the false alarm samples and found that when the container is excessively heavy, even during the normal container-lifting operation, there can be a significant upward bounce of the truck body, leading our system to incorrectly identify the operation "danger". Increasing the thresholds

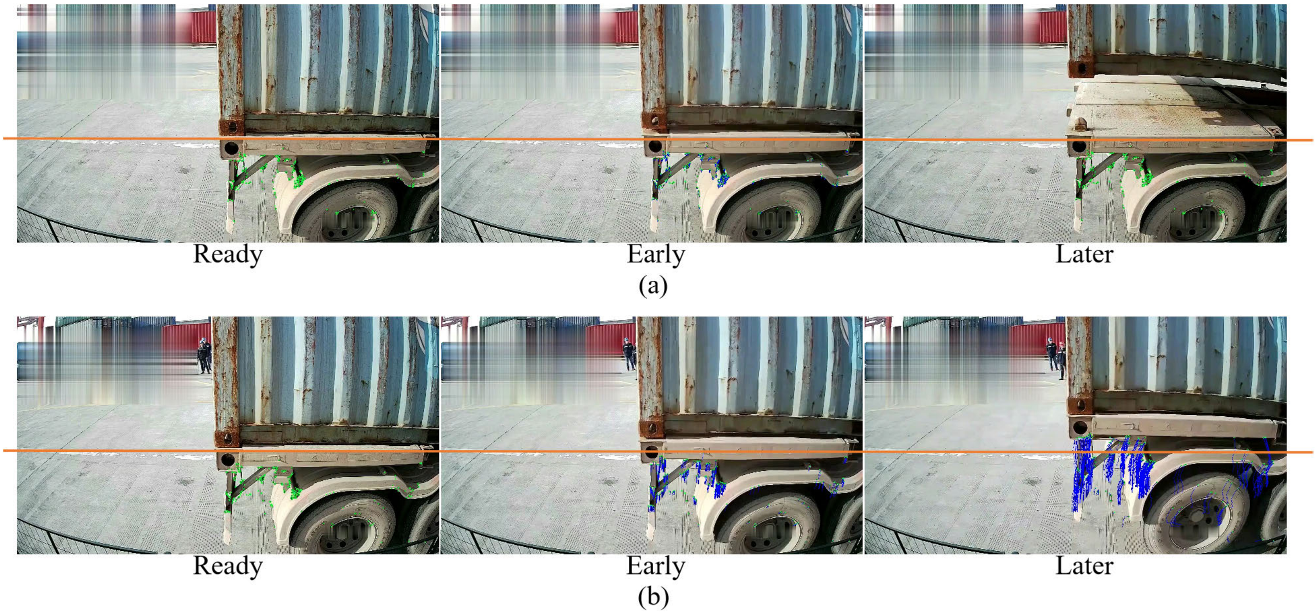


FIGURE 10. Truck-lifting detection results of our method. (a) Normal container-lifting operation. (b) Truck-lifting accident operation. The first column is the moment when the truck stops and is "Ready" for lifting. The second column is the "Early" stage of the lifting operation. The third column is the "Later" stage of the lifting operation. The green dots represent the key points extracted within the truck body, the blue curves depict the trajectories during the lifting operation, and the orange line is the reference line for the initial position of the truck body.

T_d is a potential solution. However, such an adjustment concurrently lowers the system's sensitivity and recall rate. In practical applications, it is acceptable to introduce a small number of false alarms while maintaining a high recall rate. Designing a dynamic threshold mechanism is the direction for our future research.

FIGURE 11. Key-points (green dots) within entire images. There are many useless and even interfering key-points. It is challenging to identify which object each key-point belongs to, whether the truck is present, where the truck body is, and whether the truck is lifted.

C. COMPARISON RESULTS

To analyze the performance of the proposed approach, we compare different methods for truck-lifting accident detection on the test dataset in terms of recall and false alarm, and the results are shown in Table 2.

1) KEY-POINTS-ONLY

Key-points-only method refers to directly extracting and tracking key-points on the input images. During the early stages of our research, we tried to develop the truck-lifting accident detection method only using key-points tracking but it did not work at all. This is because there are a lot of interfering objects, and we can not identify whether the truck is present, where the truck body is, and whether the truck is lifted. So the test result of this method is not provided in Table 2. We use ORB algorithm to extract key-points from the entire images, and some results are shown in Fig. 11.

TABLE 2. Comparison results of different truck-lifting accident detection methods. Detector-only method is only tested on the test set in section IV-B1, whereas our method is tested on both the test set in section IV-B1 and the additional test samples in section IV-B4.

Method	Recall	False Alarm
Key-points-only	/	/
Detector-only	56%	54%
Detector+ORB(ours)	100%	0.42%

2) DETECTOR-ONLY

In theory, only using an object detector can analyze the motion of the truck body, but the experimental results show that the boundary localization of the truck body is not accurate enough, this limitation is similar to the description mentioned in [9]. It can be seen from Table 1 that the map@0.5 is very high, but the map@0.95 is not high enough, that is, when the IoU threshold is set to 0.95, the accuracy of the detector will be greatly reduced, suggesting that detector's boundary localization accuracy falls short for reliable truck-lifting accident detection. Fig. 12 shows a representative bad case of the method using only the detector.

3) METHOD IN [9]

As mentioned in [9], their method fails when the wheel hubs are distorted or defaced (incomplete). So their method is

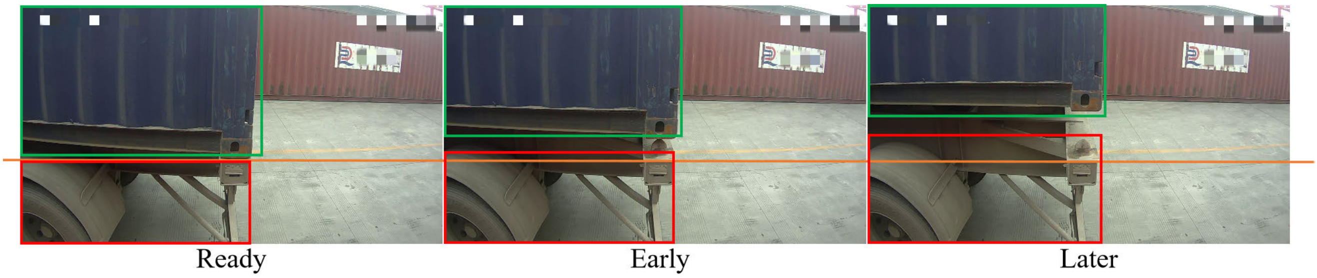


FIGURE 12. **A representative false alarm example of the detector-only method. It occurs due to jitter of the bounding box.**

FIGURE 13. **Wheel hubs detection results of the method in [9] in our scenario. The green rectangles represent the approximate regions of wheel hubs detected by the first-stage object detector in [9], while the Hough circle detector, i.e., the second-stage detector in [9], fails to further locate the wheel hubs.**

completely unsuitable for our scenario (using Hough circle detection method to detect distorted, incomplete, or even non-existent wheel hubs).

We reproduced the method in [9] to detect wheel hubs, but it did not work completely on our test set, and Fig. 13 shows some examples of failures. Even though the first-stage detector is capable of detecting distorted or incomplete wheel hubs, the second-stage Hough circle detector still fails to locate wheel hubs.

V. CONCLUSION

We have presented a novel vision-based truck-lifting accident detection method for truck-lifting prevention system in this paper. The workflow for truck-lifting accident detection is constructed based on the idea that utilizing a visual object detector to detect the truck body within which to extract some key-points whose displacements are tracked to determine whether the truck is lifted. The experiments demonstrate that our method is feasible and reliable, and is free from the disturbance of distorted, incomplete, or even unseen wheel hubs in the image, could robustly detect truck-lifting accidents with an extremely low false alarm rate. In addition, our YOLOv5-based modified detection model could reduce the computation cost of container and truck body detection and help us to reduce the cost of the whole system. Further investigation of truck-lifting accident detection with only normal samples or few negative samples will be an interesting direction, because collecting sufficient negative samples, i.e., real truck-lifting accident samples, is a dangerous and cost-consuming work.

REFERENCES

[1] W. Yan, Y. Zhu, and J. He, "Performance analysis of a new type of automated container terminal," *Int. J. Hybrid Inf. Technol.*, vol. 7, no. 2, pp. 237–248, Mar. 2014.

[2] H. Yu, Y. Deng, L. Zhang, X. Xiao, and C. Tan, "Yard operations and management in automated container terminals: A review," *Sustainability*, vol. 14, no. 6, p. 3419, Mar. 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/6/3419

[3] W. K. Kon, N. S. F. A. Rahman, R. M. Hanafiah, and S. A. Hamid, "The global trends of automated container terminal: A systematic literature review," *Maritime Bus. Rev.*, vol. 6, no. 3, pp. 206–233, Sep. 2021.

[4] N. Tsolakis, D. Zissis, S. Papaefthimiou, and N. Korfiatis, "Towards AI driven environmental sustainability: An application of automated logistics in container port terminals," *Int. J. Prod. Res.*, vol. 60, no. 14, pp. 4508–4528, Jul. 2022.

[5] C. Mi, Y. Huang, C. Fu, Z. Zhang, and O. Postolache, "Vision-based measurement: Actualities and developing trends in automated container terminals," *IEEE Instrum. Meas. Mag.*, vol. 24, no. 4, pp. 65–76, Jun. 2021.

[6] B. Brinkmann, "Operations systems of container terminals: A compendious overview," in *Handbook of Terminal Planning*, vol. 49. New York, NY, USA: Springer, Feb. 2011, pp. 25–39.

[7] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6240–6245.

[8] C. Torresan, A. Berton, F. Carotenuto, U. Chiavetta, F. Miglietta, A. Zaldei, and B. Gioli, "Development and performance assessment of a low-cost UAV laser scanner system (LasUAV)," *Remote Sens.*, vol. 10, no. 7, p. 1094, Jul. 2018.

[9] Q. Huang, Y. Huang, Z. Zhang, Y. Zhang, W. Mi, and C. Mi, "Truck-lifting prevention system based on vision tracking for container-lifting operation," *J. Adv. Transp.*, vol. 2021, pp. 1–9, Dec. 2021.

[10] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[11] G. Jocher, "Ultralytics/YOLOv5: V6.0—YOLOv5n 'nano' models, RoboFlow integration, TensorFlow export, OpenCV DNN support," Tech. Rep., 2021, doi: 10.5281/zenodo.5563715.

[12] S. Guan, M. Lei, and H. Lu, "A steel surface defect recognition algorithm based on improved deep learning network model using feature visualization and quality evaluation," *IEEE Access*, vol. 8, pp. 49885–49895, 2020.

[13] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[14] D. B. Giap, T. Ngoc Le, J.-W. Wang, and C.-N. Wang, "Wavelet subband-based tensor for smartphone physical button inspection," *IEEE Access*, vol. 9, pp. 107399–107415, 2021.

[15] H. Kawai, Y.-B. Kim, and Y. Choi, "Measurement of a container crane spreader under bad weather conditions by image restoration," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 1, pp. 35–42, Jan. 2012.

[16] C. Mi, L. Cao, Z. Zhang, Y. Feng, L. Yao, and Y. Wu, "A port container code recognition algorithm under natural conditions," *J. Coastal Res.*, vol. 103, p. 822, Jun. 2020. [Online]. Available: https://www.jstor.org/stable/48639862

[17] Z. Wang and Y. Wang, "FRCA: High-efficiency container number detection and recognition algorithm with enhanced attention," *Proc. SPIE*, vol. 11373, pp. 35–45, Jan. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:210879786

[18] C. Mi, Z. Zhang, Y. Huang, and Y. Shen, "A fast automated vision system for container corner casting recognition," *J. Mar. Sci. Technol.*, vol. 24, pp. 54–60, Feb. 2016.

[19] J. Lee, "Deep learning-assisted real-time container corner casting recognition," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 1, pp. 180–191, 2019.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, 2016, pp. 21–37.

[21] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

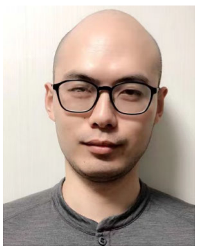
[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[23] G. Bradski, "The OpenCV library," *Dr. Dobb's J., Softw. Tools Prof. Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

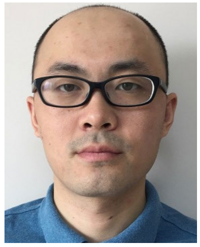
[26] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1–8.



ZHIYU JI received the master's degree from Nanjing University of Science and Technology, China. From 2014 to 2017, he was the Manager of the Engineering and Technology Department, Nanjing Port Longtan Tianchen Wharf Ltd., Nanjing, China. He was the Manager of the Production Operations Department, Nanjing Port Longtan Tianchen Wharf Ltd., from 2017 to 2018. He was the Deputy General Manager and a Party Committee Member with the Second Port Branch of Nanjing Port (Group) Ltd., from 2019 to 2021. From 2021 to 2023, he was the General Manager and the Deputy Party Secretary of Nanjing Port Longtan Container Ltd. Since 2023, he has been the Minister of Investment and Development of Nanjing Port (Group) Ltd., and the Secretary General of the Port Association. At the same time, he is currently the General Manager and the Vice Secretary of the Party Committee of Nanjing Port Longtan Container Ltd.

KAIKAI ZHAO received the B.S. degree in communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016, and the M.S. degree in communication engineering from the University of Electronic Science and Technology of China, Sichuan, China, in 2019.

In 2019, he joined CloudMinds Technologies Inc., as an Algorithm Engineer. Since 2020, he has been a Deep Learning Algorithm Engineer with the AI Innovation Center, China Unicom. His research interests include machine vision, computer vision, and deep learning.



ZHAOXIANG LIU received the B.S. and Ph.D. degrees from the College of Information and Electrical Engineering, China Agricultural University, Beijing, China, in 2006 and 2011, respectively.

He joined VIA Technologies Inc., Beijing, in 2011. From 2012 to 2016, he was a Senior Researcher with the Central Research Institute, Huawei Technologies, Beijing. He was the Senior Manager of CloudMinds Technologies Inc., Beijing, from 2016 to 2019. Since 2019, he has been the Director of AI research of the AI Innovation Center, China Unicom. He has published more than 20 refereed papers in international journals and conferences and holds more than 40 patents. His current research interests include artificial intelligence, computer vision, deep learning, robotics, and human–computer interaction.

HUAN HU received the B.S. degree from the School of Science, Hubei University of Technology, Hubei, China, in 2013, and the M.S. degree from the School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Sichuan, China, in 2017.

In 2017, he joined CloudMinds Technologies Inc., as an Algorithm Engineer. Since 2020, he has been a Deep Learning Algorithm Engineer with the AI Innovation Center, China Unicom. His research interests include machine vision, computer vision, and deep learning.

ZHONGHENG SUN received the bachelor's degree from the Kede College, Capital Normal University, China. He was the Industry Customer Manager of China United Network Communications Ltd., Nanjing Branch, from 2014 to 2021. During the period, Nanjing Port 5G+AI intelligent management platform, he participated in won the Excellence Award of the fourth "Bloom Cup" 5G Application Competition in the National Division and the third prize of the fifth "Bloom Cup" 5G Application Competition in Jiangsu Division. Since 2022, he has been the Director of the Government Business Group in China United Network Communications Ltd., Nanjing Branch. He has led many important benchmark projects of Nanjing Public Security, such as the intelligent vehicle control platform, which uses artificial intelligence to automatically analyze and control the illegal state of vehicles. Meanwhile, he participated in Nanjing Port 5G+AI intelligent management platform, which won the honor of credible AI case and an excellent case of artificial intelligence platform application.

SHIGUO LIAN (Member, IEEE) received the Ph.D. degree from Nanjing University of Science and Technology, China.

He was a Research Assistant with the City University of Hong Kong, in 2004. From 2005 to 2010, he was a Research Scientist with France Telecom Research and Development Beijing. He was a Senior Research Scientist and the Technical Director of the Huawei Central Research Institute, from 2010 to 2016. He was the Senior Director of CloudMinds Technologies Inc., from 2016 to 2019. Since 2019, he has been the Chief AI Scientist of China Unicom. He has authored more than 100 refereed international journal articles covering topics of artificial intelligence, robotics, human–computer interface, and multimedia communication. He has authored or coedited more than ten books and holds more than 200 patents. He currently serves on the editor board for several refereed international journals.