WDCNN-BASED FAULT DIAGNOSIS METHOD FOR AGV ON CLOUD PLATFORM

Zhi Qiu,[*] Zeyu Xu,[*] Min Luo,[*] Simon X. Yang,[**] and Tao Chen[*]

Abstract

Automated-guided vehicles (AGVs) are widely used in the manufacturing and logistics industry, so it is crucial to meet the demand for intelligent, efficient, and accurate fault diagnosis of AGVs. This study proposes a fault diagnosis method based on deep convolutional neural networks with a wide first-layer kernel (WDCNN), which enables a more complete extraction of fault features. A cloud platform trolley data acquisition system and an experimental AGV platform are established. Compared to traditional machine learning methods, WDCNN achieves more accurate differentiation and localisation of faults in the same part and obtains a higher diagnostic accuracy of over 99%. With the proposed method, the results of cloud-based diagnosis exceed 98% accuracy, micro-F1-scores surpass 0.98, and the prediction time is within 0.009 s. The proposed method is suitable for fault diagnosis of AGV trolleys under remote and unmanned monitoring.

Key Words

AGV fault diagnosis; convolution neural network; cloud platform

1. Introduction

In the era of Industry 4.0, high-level information and communication technologies, such as artificial intelligence, the Internet of Things, big data, and cloud systems, could become highly integrated with society. As such, logistics and manufacturing industries are booming, and factories are being increasingly automated. Automated-guided vehicles (AGVs) are poised to become some of the most important types of automation equipment [1], but their safety remains to be optimised [2].

Currently, most studies on AGVs focus on path planning [3], scheduling problems [4], and body system design [5], and there are few articles on the fault diagnosis of AGVs [6], [7] proposed a virtual sensor-based fault diagnosis method considering the cost and battery capacity of AGVs. The method constructs a compact dynamics model of the AGV cart without the need to determine the cart tire model, generating residuals with the output estimated by the virtual sensor. The validity and precision of the diagnosis method were experimentally demonstrated. [8] presents a dynamic model that can predict the wheel speed of a robot and use the residual signals of the predicted and measured states to detect faults in mobile robots. [9] mathematically modelled the AGV system, used multiple positioning modules to obtain the position information of the AGV, and implemented an extended Kalman filter to detect errors.

However, fault diagnosis methods based on modelling analysis still have drawbacks. For example, as the structure of AGV trolleys becomes more complex, it becomes more difficult to construct accurate and generalisable AGV models [10]. Furthermore, the extraction of fault features requires *a priori* knowledge, diagnostic experience, or expertise accumulated over time [11]. Traditional feature extraction methods can be quite complex, and different features are suitable for different signals. Existing methods may not be able to fully reflect fault features or achieve real-time fault detection [12]. In [13], four types of machine learning (RNN, SVM, RF, and ANN) were used to detect faults in the actuators and sensors of AGV.

Data-driven deep learning methods have attracted much attention due to their intelligent and efficient modes [14]–[16] For IMU deviation and drift faults, a dynamic DBN model design with inexact LSA-GA-based weight value optimisation for fault diagnosis was employed. Ding *et al.* [17] used a decision tree model to preclassify data into normal, abnormal, and uncertain states, and then used long short-term memory (LSTM) networks to perform fault diagnosis on the uncertain state data. Wang *et al.* [18] transformed the vibration signals of AGV core components into two-dimensional images and used VGGNet for fault diagnosis. Ding *et al.* [19] constructed a twin-tower model for fault detection, composed of attention networks for learning key variables in sensor data and LSTM networks for learning event dimensional features. Miao *et al.* [20] proposed a novel channel-wise convolutional neural network (CNN) with feature enhancement for

[*] Southwest Petroleum University, Chengdu, China; e-mail: yallym@163.com; {984182621, 36205007, 1394174885}@qq.com
[**] University of Guelph, Guelph, ON, Canada; e-mail: syang@uoguelph.ca
Corresponding author: Zhi Qiu

Recommended by Stephen Andrew Gadsden
(DOI: 10.2316/J.2024.206-0952)

fault diagnosis of wheeled mobile robots. Important features were highlighted by adaptive weighting, and a channel convolution mechanism was introduced to prevent these features from being mixed [21]. A spatial–temporal difference graph convolutional network (STDGCN) was proposed for fault diagnosis of wheeled robots. It uses the graph-structured data along with the association graph as input. Experimental results demonstrate the excellent performance of STDGCN [22]. A data fusion method based on CNN (LeNet-5) was proposed. By performing Fourier transform on the sound and vibration signals of the left and right motors separately, frequency spectrograms were obtained. After data fusion, they were fed into the LeNet-5 model. Compared to single-sensor input and dual-sensor input, this method yielded better diagnosis results. These studies have advanced data-driven fault diagnosis methods in the field of AGVs, but the use of two-dimensional images for fault diagnosis ignores the translational invariance of the periodic signals and only locates faults to different parts of the AGV in a general way without segmenting the faults.

To eliminate these shortcomings, this paper proposes an AGV fault diagnosis method based on a cloud platform and deep CNNs with a wide first-layer kernel (WDCNN). Without human intervention, the AGV uploads sensor data to the cloud platform, and the trained WDCNN uses the cloud platform data as input for trolley fault diagnosis. Several sets of comparative experiments are conducted based on the fault diagnosis dataset recorded by the Mecanum wheel AGV. In addition, t-SNE technology [23], [24] is used to visually demonstrate the ability of different networks in fault feature learning and classification. It maximises the similarity of adjacent data points to preserve the relative relationships between the data, providing a new method for better visualisation. It can help us discover hidden structures and clusters in the data set, facilitating more in-depth analysis of the data. The results demonstrate that WDCNN can extract fault features autonomously, and the wide convolution kernel mechanism enables the network to get a larger perceptual field to better extract fault features for satisfactory diagnostic performance. The main contributions of this study are summarised as follows.

(1) Introducing a wide convolutional kernel mechanism into the 1DCNN network expands the perceptual field of the convolutional kernel and enables the network to extract more complete fault feature information.

(2) Introducing a cloud platform solves the contradiction between the limited computation and storage resources of AGV trolley ontology and the high computation and memory capacity requirements of complex fault diagnosis algorithms, which can reduce the cost and promote the industrialisation of AGV trolleys.

(3) Experimental results show that the proposed network with WDCNN and cloud platform can realise the online fault diagnosis of AGV trolleys and meet the intelligence and safety requirements of future manufacturing and logistics industries.

2. Basic Architecture of WDCNN

2.1 Convolutional Layer

The CNN (ConvNet) is a deep feedforward neural network with weight sharing and local connectivity features [25]. The network in this paper uses a broad convolutional kernel [26] with the following convolutional expression:

$$y_{l+1,m}\left(n\right) = w_{l,m} *\ x_l\left(n\right) + b_{l,m}, \qquad (1)$$

where $w_{l,m}$ is the weight matrix of the lth filter in the mth layer; $b_{l,m}$ is the bias term; $x_l\left(n\right)$ is the nth input; and $y_{l+1}\left(n\right)$ is the nth output after convolution with the mth filter.

2.2 Activation Layer

The activation layer improves the nonlinearity of the output after convolution, thus enhancing the feature expression of the model. This paper uses ReLU as the activation function to accelerate the convergence of CNN while improving the sparsity of the network and effectively avoiding the overfitting problem, and it is expressed as:

$$a_{l+1,m}\left(n\right) = f\left(y_{l+1,m}\left(n\right)\right) = \max\left\{0\ ,\ y_{l+1,m}\left(n\right)\right\}, \qquad (2)$$

where $a_{l+1,m}\left(n\right)$ is the output value after $y_{l+1,m}\left(n\right)$ activation, and $f\left(\cdot\right)$ is the activation function.

2.3 Pooling Layer

The role of pooling is to reduce dimensionality. When the input data are slightly skewed, the pooling returns the same output as before regardless of the slight deviation, so it is robust to the input data and can prevent overfitting. This paper uses Max Pooling, which can better reflect the periodic characteristics of the fault. It is expressed as follows:

$$p_{l+1,m}\left(n\right) = \max_{(n-1)H+1\leq i\leq nH}\left\{q_{l,m}\left(i\right)\right\}, \qquad (3)$$

where $q_{l,m}\left(i\right)$ is the value of the lth neuron in the mth filter in the ith layer, $i \in \left[\left(n-1\right)H+1, \mathrm{nH}\right]$, H is the width of the pooling area, and $p_{l+1,m}\left(n\right)$ is the output value after the pooling operation.

2.4 Training of the WDCNN

For training in WDCNN, the loss function is chosen to be categorical cross-entropy with the following expression:

$$\mathrm{Loss} = -\frac{1}{S}\sum_{s=1}^{S}\left[y\ln a + \left(1-y\right)\ln\left(1-a\right)\right] + \mathrm{L2}, \qquad (4)$$

where S is the number of samples, y is the true label of the corresponding sample, and a is the true output of the neuron. $L2$ is the regularisation term, which is used to prevent the overfitting problem, and its expression is as follows:

$$\mathrm{L2} = \frac{\lambda}{2S}\sum_{l=1}^{L}\left\|w^l\right\|^2, \left\|w^l\right\|^2 = \sum_{j=1}^{N_l}\sum_{k=1}^{N_{l+1}}\left(w_{k,j}^l\right)^2, \tag{5}$$

where $w_{k,j}^l$ is the lth layer weight of the network, which connects the kth and jth neurons of the latter layer, and λ is the regularisation factor.

The weights and biases of the network are updated using the Adam optimiser. The method calculates the adaptive learning rate for different parameters from the budget of the first moment m_t and the second moment r_t of the gradient. The updating process can be expressed as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, r_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \tag{6}$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}, \widehat{v_t} = \frac{v_t}{1-\beta_2^t}, \tag{7}$$

$$\theta_t = \theta_{t-1} - \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}}+\delta}\mathrm{lr}, \tag{8}$$

where g_t is the gradient, β_1 and β_2 are the decay rates, δ is a constant that can be ignored, and lr is the learning rate.

2.5 Hyperparameter Preselection

The core idea of hyperparameter preselection is to select a large perceptual field to capture more fault information and extract less invalid information.

Given that a wheel rotates periodically and the phase value of each input signal is different, to avoid CNN learning, the unit neuron in the final pooling layer of the network is expected to extract at least one cycle of the wheel speed signal. Therefore, the following equation must be satisfied:

$$f \cdot \frac{60}{r} \leq V^0 \leq L, \tag{9}$$

where V^0 is the receptive field of the neurons in the final pooling layer, f is the sampling frequency, r is the minimum wheel speed, and L is the length of the input sample sampling points. In the experiments of this paper, the minimum wheel speed is 40 rpm and the sampling frequency is 100 Hz, so the sensory field of the first convolutional layer should be larger than 150 sampling points.

The relationship between the nth pooling layer and the $n-1$th pooling layer in the network can be expressed as follows:

$$V^{(l-1)} = S^l\left(P^l \cdot V^l - 1\right) + w^l, \tag{10}$$

where S^l represents the step size of the lth convolutional layer, w^lrepresents the convolutional kernel width of the lth convolutional layer, and P^lrepresents the pooling kernel width of the lth pooling layer; in the pre-defined model structure, when $l > 1$, $S^l = 1$, $W^l = 3$, and $P^l = 2$. Therefore, (10) can be rewritten as follows:

$$V^{l-1} = 2V^l + 2. \tag{11}$$

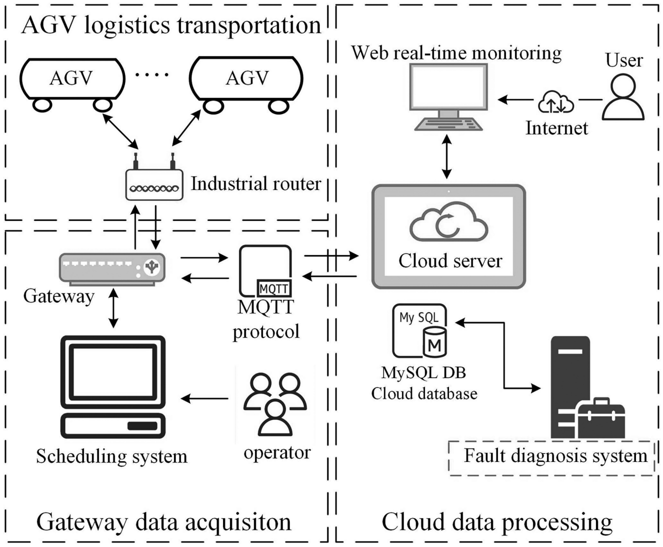


Figure 1. System architecture diagram.

The formula for mapping the final pooling layer, that is, the nth pooling layer unit neuron sensory field, to the first pooling layer is

$$V^1 = 2^{n-1} \times 3 - 2. \tag{12}$$

Substituting (12) into (11) yields the receptive field expression for the final layer unit neuron on the input signal:

$$V^0 = S^1\left(P^l \cdot \left(2^{n-1} \times 3 - 2\right) - 1\right) + w^1 = S^1\left(2^n \times 3 - 4\right) + w^1 - 1. \tag{13}$$

According to the two prerequisites that $f \cdot \frac{60}{r} \leq V^0 \leq L$ and S^1 need to be divisible by L, in the case of convolution by five layers, the step size of the first convolution layer can only be 2, 4, 8, or 16. When reducing the stride, the number of trainable neurons in the network will increase, which can improve the network's ability to learn fault features. However, it also increases the risk of model overfitting and the training time required for the network. Therefore, the first convolutional layer is selected with a stride of 16 and a kernel size of 64, with the specific parameters shown in Table 1.

3. Cloud Platform Construction and AGV Fault Diagnosis Process

3.1 Overall System Architecture

The system is built on IoT and industrial cloud platform technologies, mainly consisting of three parts: the AGV data collection and transmission platform, the gateway data forwarding, and the cloud data processing. The architecture of the system is shown in Fig. 1.

The AGV data collection platform uses a built-in wireless WiFi module to upload data from various sensors. The gateway data forwarding is centered on an intelligent gateway that uses lightweight network

Table 1
Network Topology

Network Structure	Kernel Size	Strides	Filters	Output	Padding
Convolutional layer	64	16	16	128*16	Same
Pooling layer	2	1	16	64*16	Valid
Convolutional layer	3	32	32	64*32	Same
Pooling layer	2	1	32	32*32	Valid
Convolutional layer	3	64	64	32*64	Same
Pooling layer	2	1	64	16*64	Valid
Convolutional layer	3	64	64	16*64	Same
Pooling layer	2	1	64	8*64	Valid
Convolutional layer	3	64	64	6*64	Same
Pooling layer	2	1	64	3*64	Same
Dense	–	–	–	192	–
Softmax	–	–	–	8	–

Table 2
Advantages and Disadvantages of Different Protocols

Protocol Type	Advantages	Disadvantages
HTTP	HTTP protocol is widely used in the Internet, and its technology is mature and easy to develop.	HTTP protocol is one-way and cannot passively receive commands from the network.
CoAP	CoAP protocol is an application layer protocol built on top of UDP protocol, mainly used in lightweight M2M communication.	The security of data cannot be guaranteed.
NB-IoT	NB-IoT protocol is built and operated on cellular networks, consumes low bandwidth, and can be directly deployed on existing GSM or LTE networks.	Low data transmission and high communication costs.
LoRaWAN	LoRaWAN protocol has the characteristics of long-distance use and low power consumption.	The security of data cannot be guaranteed and is susceptible to attacks.

management to enable communication between the robot, scheduling system, and cloud. Additionally, to ensure the correctness and security of data exchange between the gateway and the cloud, communication protocols need to be established between devices. Common IoT communication protocols include HTTP, MQTT, CoAP, NB-IoT, and LoRaWAN, and their advantages and disadvantages are compared in Table 2. The MQTT protocol, based on a publish/subscribe model, has a higher tolerance for unstable networks and faster response times and larger throughput than other protocols, making it ideal for cloud data interaction. Cloud data processing leverages the high virtualisation and powerful computing capabilities of cloud platforms to replace traditional fault platforms embedded with upper computer software, effectively improving the real time and convenience of fault diagnosis.

Remote AGV status monitoring and online fault diagnosis are completed by extracting the data from the database. The fault diagnosis process can be divided into two parts: model training and online diagnosis. The detailed flowchart of AGV fault diagnosis is shown in Fig. 2.

In the model training part, the trolley is made to upload fault data by artificially creating faults. After collecting a sufficient amount of AGV faulty wheel speed signals, the fault dataset is constructed using the sliding-window method [27]. The network is trained using the training set, so the CNN layer achieves feature information and updates the model parameters using the backpropagation algorithm. After completing the model training, the online diagnosis part can be performed. The model diagnoses the status of the AGV using the sensor data uploaded in real time. The result of the diagnosis is

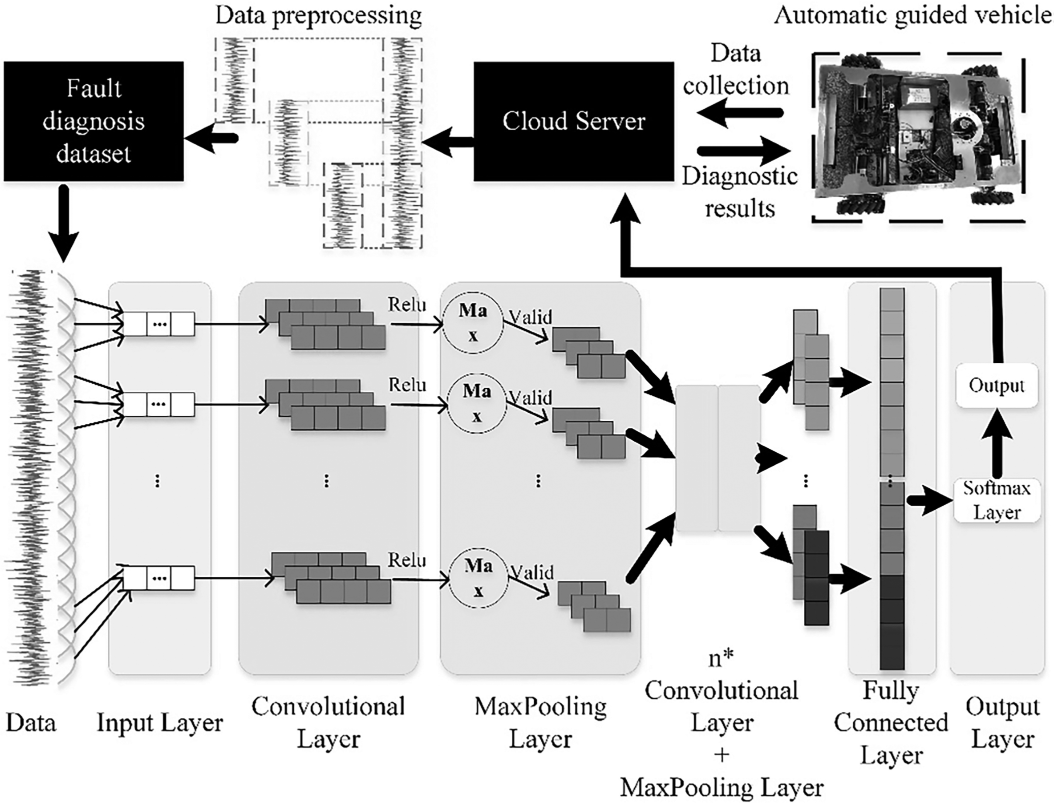


Figure 2. AGVs fault diagnosis flowchart.

fed back to the cloud database, and the intelligent gateway judges the fault level and decides the next action of the AGV, ensuring the safety of the AGV while improving the efficiency of the plant.

4. Experimental Demonstration

4.1 Introduction of the Experimental Platform

To verify the effectiveness of the proposed method for the diagnosis of AGV cart faults, an AGV is built as an experimental platform, and its hardware structure diagram is shown in Fig. 3. The cart realises the navigation and positioning of the AGV through the magnetic navigation sensor and the RFID reader. The AGV is also controlled by STM32 with four Mecanum wheels, and equipped with a DJI RM35 DC motor, an incremental encoder, and a nine-axis gyroscope (IMU).

The CPU model of the computer used for the experiments was Intel(R) Core(TM) i3-7100 CPU@ 3.90GHz with 16GB of RAM. The programming language was Python, using the TensorFlow framework developed by the Google Brain team, version 2.3.0, and keras version 2.4.3. To eliminate the effect of randomness on the experimental results, each round of experiments was carried out ten times and the average value was calculated.

4.2 Data Collection and Processing

Assuming that the trolley is running in a straight line, mechanical and operational faults of the trolley wheels are created by manual simulation. For the power system failure, the setup is as shown in Fig. 4. In Fig. 4(a), the red circle shows that tape is wrapped around one of the small wheels in the Mecanum wheel, limiting the wheel's rotation, to simulate a possible wheel locking failure that may occur (state code 06, 07). In Fig. 4(b), the red circle shows that one of the small wheels in the Mecanum wheel is removed to simulate a wheel broken failure (state code 00). In Fig. 4(c), the axes of the motor and the wheel are

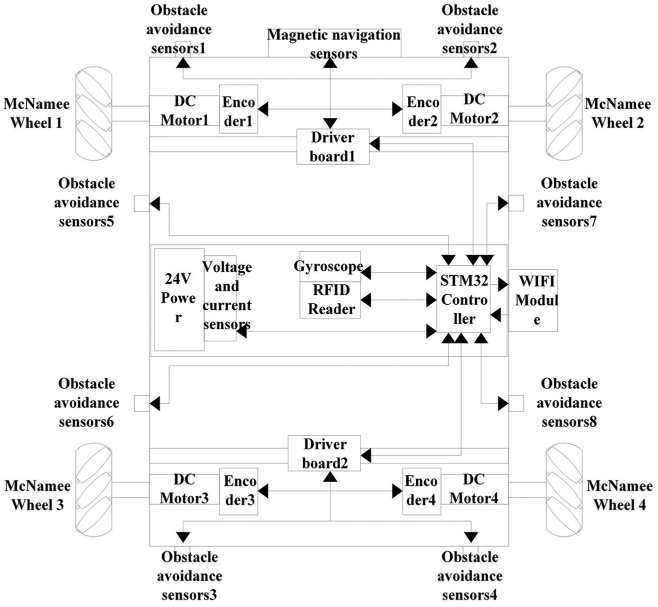

Obstacle avoidance sensors1
Magnetic navigation sensors
Obstacle avoidance sensors2
McNamee Wheel 1
DC Motor1
Encoder1
Encoder2
DC Motor2
McNamee Wheel 2
Driver board1
Obstacle avoidance sensors5
Obstacle avoidance sensors7
24V Power
Voltage and current sensors
Gyroscope
RFID Reader
STM32 Controller
WIFI Module
Obstacle avoidance sensors6
Obstacle avoidance sensors8
Driver board2
McNamee Wheel 3
DC Motor3
Encoder3
Encoder4
DC Motor4
McNamee Wheel 4
Obstacle avoidance sensors3
Obstacle avoidance sensors4

Figure 3. Magnetically-guided AGVs with Mecanum wheels.

Table 3
AGV Failure Dataset Division

Code	AGV Status	Sample size		
		Training Set	Validation Set	Test Set
00	Left front wheel broken	2100	600	300
01	Motor fault	2100	600	300
02	Encoder fault	2100	600	300
03	Left front wheel normal	2100	600	300
04	IMU normal	2100	600	300
05	IMU fault	2100	600	300
06	Left front wheel locked	2100	600	300
07	Right rear wheel locked	2100	600	300

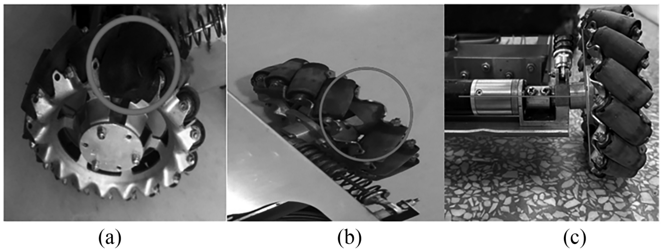


Figure 4. Manually set faults: (a) wheel lock; (b) wheel breakage; and (c) motor fault.

not perpendicular to the cross-section of the wheel as they should be, simulating a motor shaft offset failure caused by collision (state code 01). It is difficult to display sensor failures (state codes 02, 04, 05) in pictures and they are not listed here.

The wheel speed signal of the AGV is collected by the encoder with a sampling frequency of 100 Hz. The datasets are described in Table 3. Because a large amount of data is required to train a deep-learning network, this paper uses sliding window sampling to prevent overfitting. There are two parameters for the sliding window sampling method: window size and sliding window steps. The window size determines the number of points sampled in each sample. A change in the sliding window step size affects the fault features contained in the samples and may affect the fault diagnosis of the model. In this paper, we choose a window size of 2048 and a sliding window step size of 50. Due to the use of sliding window sampling, there are duplicate data between adjacent samples. To avoid leakage of the test set during model training, The training set has no intersection with other sets.

4.3 Validation Setup

The evaluation metrics are based on the confusion matrix made up of the model diagnosis results and the true labels. Samples can be classified into four discriminatory results: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The metrics used in this paper are precision (P), recall (R), F1-score, and accuracy (A), which are defined as follows:

$$P = TP/\left(TP + FP\right) \times 100\%, \tag{14}$$

$$R = TP/\left(TP + FN\right) \times 100\%, \tag{15}$$

$$\begin{cases} \text{micro} - P = \frac{\overline{\text{TP}}}{\overline{\text{TP}}+\overline{\text{FP}}} \\ \text{micro} - R = \frac{\overline{\text{TP}}}{\overline{\text{TP}}+\overline{\text{FN}}} \\ \text{micro} - F1 = \frac{2\times\text{micro}-P\times\text{micro}-R}{\text{micro}-P+\text{micro}-R} \times 100\% \end{cases}, \tag{16}$$

$$A = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%. \tag{17}$$

4.4 Comparison With Other Approaches

To verify the superiority of the proposed model for the diagnosis of AGV cartwheel failures, it is compared with other traditional machine learning algorithm models. Using Keras framework, the models of RNN [28], BiLSTM [29], GRU [30], LeNet [31], AlexNet, and ResNet are built and validated under the same dataset Table 3, and the experimental results of each model are shown in Table 4.

As can be seen in Table 4, the F1-scores of the RNN and LeNet networks are only 0.9068 and 0.9288, respectively. Due to their simple structures and insufficient network depth, these two networks are difficult to accurately diagnose faults. The F1-score of BiLSTM is 0.9599, which introduces gate mechanisms and has the ability to remember long-term and short-term information, enabling it to better capture bidirectional feature dependencies and effectively solve the problems of gradient disappearance and explosion in RNN, but its ability to capture fault feature information is still weak. GRU also introduces gate mechanisms, has fewer parameters, faster convergence speed, and less time consumption, accelerating the iteration process. Its F1-score is 0.9687. By increasing the number of network layers, the F1-score of AlexNet is 0.9815, and that of ResNet is 0.9683, which is significantly improved

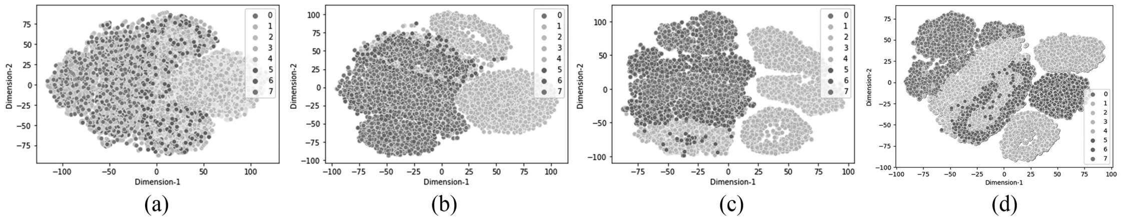


Figure 5. Release of features after t-SNE dimensionality reduction: (a) distribution of original fault characteristics; (b) RNN t-SNE; (c) BiLSTM t-SNE; and (d) GRU t-SNE.

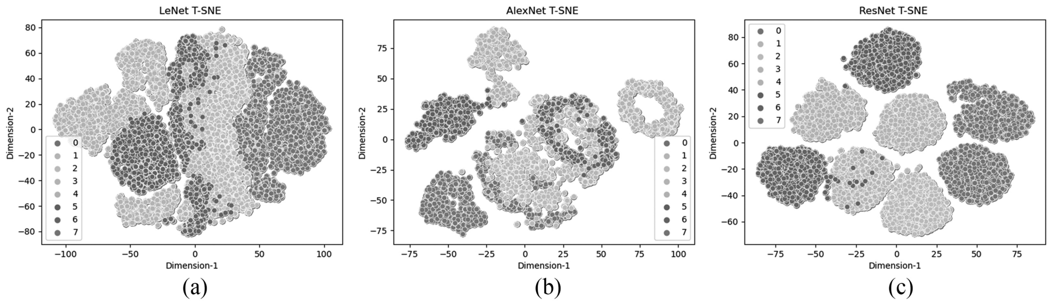


Figure 6. Release of features after t-SNE dimensionality reduction: (a) LeNet t-SNE; (b) AlexNet t-SNE; and (c) ResNet t-SNE.

Table 4
Fault Classification Results of Different Models

Model	Precision	Recall	F1-score	Accuracy
RNN	0.9113	0.9079	0.9068	0.9068
BiLSTM	0.9608	0.9592	0.9599	0.9593
GRU	0.9694	0.9688	0.9687	0.9688
LeNet	0.9332	0.9283	0.9288	0.9283
AlexNet	0.9817	0.9817	0.9816	0.9817
ResNet	0.9711	0.9683	0.9683	0.9683
WDCNN	0.9983	0.9983	0.9983	0.9983

compared to LeNet. However, the network topology structure with small convolution kernels and multiple convolution channels still has limited capability to extract features from periodic signals. The proposed WDCNN network has the best diagnostic performance, with the highest scores in all diagnostic criteria, and the F1-score is at least 0.11 higher than other methods.

By using t-SNE technology, high-dimensional data can be transformed into two-dimensional features for visualisation. By visualising the original fault features and the features of the last layer of RNN, BiLSTM, and GRU networks trained on sequence data, we can reflect whether the network has learned to separate fault features. Fig. 5(a) shows the distribution of original fault data samples in two-dimensional space. It can be seen that the unclassified samples are disorderly and difficult to follow. Fig. 5(b) shows the distribution of fault features classified by RNN, and the features classified by RNN are still mixed and overlapped, and the classification effect is not ideal. In Fig. 5(c) and 5(d), BiLSTM and GRU can classify some of the fault feature signals, and the performance is better than that of RNN, but there are still some overlapping and crossing phenomena.

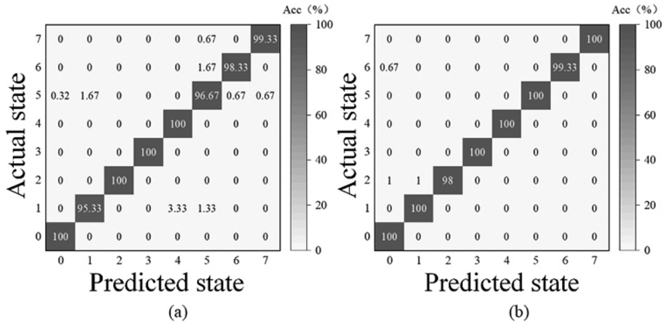
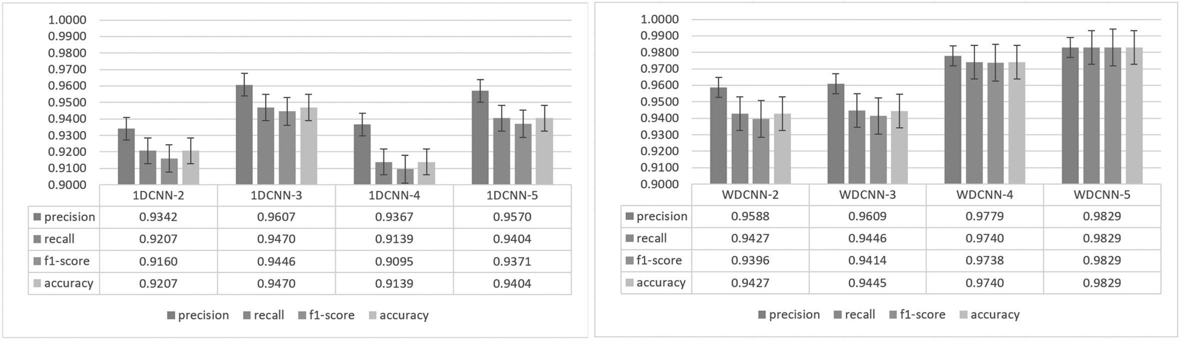


Figure 7. Confusion matrices of: (a) AlexNet and (b) WDCNN.

The t-SNE visualisations of the classification results of the LeNet, AlexNet, and ResNet models constructed based on CNNs are shown in Fig. 6. Comparing with Fig. 5, it can be seen that the convolutional architecture is more conducive to extract AGV fault features than using sequence data as input. Among them, AlexNet with a deeper architecture has better classification effect than LeNet, with tighter clustering of each fault. ResNet, using residual structure and deeper architecture without the problem of gradient disappearance, has the best classification effect. However, there are still some overlapping and crossing phenomena in some fault features.

To further analyse the diagnostic ability of the networks, the diagnostic results of the AlexNet and WDCNN networks on the test set are plotted as a confusion matrix, as shown in Fig. 7. The horizontal axis in the figure represents the predicted AGV state by



	1DCNN-2	1DCNN-3	1DCNN-4	1DCNN-5
■ precision	0.9342	0.9607	0.9367	0.9570
■ recall	0.9207	0.9470	0.9139	0.9404
■ f1-score	0.9160	0.9446	0.9095	0.9371
■ accuracy	0.9207	0.9470	0.9139	0.9404

	WDCNN-2	WDCNN-3	WDCNN-4	WDCNN-5
■ precision	0.9588	0.9609	0.9779	0.9829
■ recall	0.9427	0.9446	0.9740	0.9829
■ f1-score	0.9396	0.9414	0.9738	0.9829
■ accuracy	0.9427	0.9445	0.9740	0.9829

(a)

(b)

Figure 8. Results of different: (a) 1DCNN and (b) WDCNN networks.

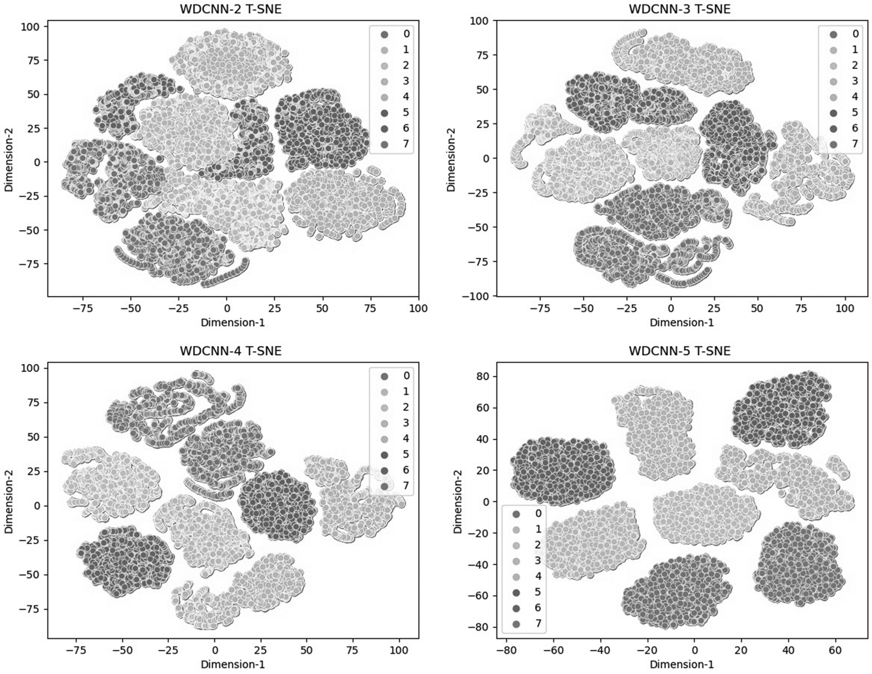


Figure 9. Characteristic output of each convolution layer after dimension reduction by t-SNE.

the model, and the vertical axis represents the actual state of the AGV. The darker the color in the bar chart on the right side, the more samples were correctly classified.

From Fig. 7(a), it can be seen that AlexNet can roughly locate and classify faults accurately, and four types of faults are correctly identified with 100% accuracy. However, due to its insufficient fault feature extraction ability, six motor failure samples (state code 01) were predicted as IMU failure (state code 05) and 13 IMU failure samples (state code 05) were predicted as wheel 1 locking (state code 06). It is difficult to classify faults accurately among different parts. From Fig. 7(b), it can be seen that six states are correctly classified with 100% accuracy. Only two samples of wheel 1 locking (state code 06) were diagnosed as wheel damage, and six encoder failure samples (state code 02) were diagnosed as wheel damage (state code 00) and motor failure (state code 01) respectively. This may be because these three types of fault features are all included in one wheel encoder. The overall accuracy of WDCNN reached 99.83%, and the diagnostic effect was good, and the precision requirements were also met.

4.5 Ablation Experiments

To study the influence of network depth and verify the effectiveness of the wide convolution kernel mechanism, eight deep learning networks are established, namely, 1DCNN-2, 1DCNN-3, 1DCNN-4, 1DCNN-5, WDCNN-2, WDCNN-3, WDCNN-4, and WDCNN-5. In addition to the width of the convolution kernel, the hyperparameter settings of each network are the same. The average diagnosis results of the models are shown in Fig. 8.

According to Fig. 8, it can be seen that when the network layer is shallow, the performance of WDCNN is similar to that of 1DCNN. However, as the network layer increases, the performance of WDCNN on the four indicators of Precision, Recall, F1-score, and Accuracy far exceeds that of 1DCNN with the same number of layers. In addition, the diagnostic results of 1DCNN

Table 5
Cloud Diagnostic Results at Different Sliding Window Step Sizes

Sliding Window Steps	Loss	Acc	Diagnosis Time (s)	F1-score
100	0.0409	0.9875	1.13	0.9875
200	0.028	1	1.96	1
300	0.041	0.9937	1.33	0.9938
400	0.047	0.9875	1.96	0.9875

at different layers fluctuate significantly, and increasing the depth of the network model does not necessarily improve its accuracy. This suggests that the approach of reducing the training parameters of the network while improving the feature extraction ability by repeating and stacking small filters, which is suitable for the field of machine vision, is not applicable to AGV fault diagnosis. With the introduction of wide convolution kernel mechanism, the network extracts more fault feature information. As the network layer deepens, the model's expressive power is further enhanced, and the accuracy is improved.

Next, t-SNE technology is used to observe the output of WDCNN convolutions at different depths. Fig. 9 shows the output of the second, third, fourth, and fifth convolutional layers of WDCNN. It can be seen that as the network layer increases, the same fault features are more tightly clustered. Compared with the classification results of LeNet, AlexNet, and ResNet in Fig. 6, which are also CNN architectures, the different fault features are separated from each other, and the effect of learning fault features is better.

4.6 Case Study

While changing the load of the AGV trolley, the trained model is used to diagnose the fault data uploaded to the SQL cloud database.

To verify the robustness of the model and simulate the randomness when faults occur, experiments are carried out in different sliding window steps, the fault characteristics contained in different steps are different, and step sizes of 100, 200, 300, and 400 are tested. Twenty sets of data for each state are collected to ensure the accuracy of the experiment. The cloud diagnostic results for each sliding window step size are shown in Table 5.

The experiments prove that the proposed fault diagnosis model based on the cloud platform can accurately identify different faults while changing the load and sliding window step size. The model's overall accuracy rate is above 98%, time to diagnose 180 sets of data is around 1.6 s, and F1-scores are all above 0.98. The robustness, accuracy, and real-time operation of the fault diagnosis method are verified.

5. Conclusions

Compared to other methods, the proposed one-dimensional wide-convolutional kernel neural network was superior in extracting fault features, establishing diagnoses from fault data, and diagnosing different faults of the same component, with an overall accuracy rate of 99%. The accuracy of online diagnosis surpassed 98% while changing the load and sliding window step, and the micro-F1-scores were all above 0.98.

The network was trained with few parameters and the online diagnosis time was short, with an average diagnosis time of 1.6 s for 180 data sets. The cloud-based fault diagnosis method showed great potential to achieve real-time, intelligent, and unmanned monitoring in the cloud, while guaranteeing the safety of AGVs and improving factory efficiency. The real-time fault diagnosis of the cloud platform was affected by the network conditions, and we expect the growing popularity of 5G technology to provide a solution.

References

[1] L. Lynch, T. Newe, J. Clifford, J. Coleman, J. Walsh, and D. Toal, Automated ground vehicle (AGV) and sensor technologies-a review, *Proc. of the* 12th *International Conf. on Sensing Technology (ICST), Limerick,* 2018, 347–52.

[2] G.K. Fourlas, G.C. Karras, and K.J. Kyriakopoulos, Sensors fault diagnosis in autonomous mobile robots using observer — Based technique, *Proc. of the International Conf. on Control, Automation and Robotics*, Singapore, 2015, 49–54.

[3] X. Wang, Z. Xia, X. Zhou, J. Wei, X. Gu, and H. Yan, Collision-free path planning for arc welding robot based on IDA-de algorithm, *International Journal of Robotics and Automation, 37*(6), 2022, 476–85.

[4] Y. Liu, D. Han, L. Wang, and C.-Z. Xu, Novel topological relationship solutions to the ALV multi-indegree–multi-outdegree task sequence planning problem, *International Journal of Robotics & Automation, 37*(3), 2022, 257–65.

[5] X. Wu, Y. Li, H. Jia, G. Ma, and Y. Zhang, Multi-step optimal predictive control for path correction of the AGV driven by hub motors, *International Journal of Robotics & Automation, 36*(5), 2021, 304–15.

[6] E. Khalastchi and M. Kalech, Fault detection and diagnosis in multi-robot systems: A survey, *Sensors (Basel), 19*(18), 2019, 4019, doi: 10.3390/s19184019.

[7] M. Witczak, M. Mrugalski, M. Pazera, and N. Kukurowski, Fault diagnosis of an automated guided vehicle with torque and motion forces estimation: A case study, *ISA Transactions, 104*, 2020, 370–81.

[8] O.N. Şahin and M.İ.C. Dede, Model-based detection and isolation of the wheel slippage and actuator faults of a holonomic mobile robot, *Industrial Robot*: *The International Journal of Robotics Research and Application*, (ahead-of-print), 2022.

[9] P.S. Pratama, A.V. Gulakari, Y.D. Setiawan, D.H. Kim, H.K. Kim, and S.B. Kim, Trajectory tracking and fault detection algorithm for automatic guided vehicle based on multiple positioning modules, *International Journal of Control, Automation and Systems, 14*(2), 2016, 400–10.

[10] *Introduction to mobile robot control.* (Amsterdam: Elsevier, 2013).

[11] F. Jia, Y.G. Lei, J. Lin , X. Zhou, and N. Lu, Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive

data, *Mechanical Systems and Signal Processing*, *72–73*, 2016, 303-15.

[12] D. Zhuo-Hua, C. Zi-Xing, and Y. Jin-Xia, Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: A survey, *Proc. of the IEEE International Conf. on Robotics and Automation*, Barcelona, 2005, 3428–33.

[13] F. Ibrahim, B. Boussaid, and M.N. Abdelkrim, Fault detection in wheeled mobile robot based machine learning, *Proc. of the 19th International Multi-Conf.* on Systems, *Signals & Devices (SSD)*, Sétif, 2022, 58–63.

[14] K. Choi, J. Yi, C. Park, and S. Yoon, Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines, *IEEE Access*, *9*, 2021, 120043–65.

[15] S. Khan and T. Yairi, A review on the application of deep learning in system health management, *Mechanical Systems and Signal Processing*, *107*, 2018, 241–65.

[16] L. Xia, S. Zhang, R. Shen, and J. Cui, Data association-based fault diagnosis of IMUs: Optimized DBN design and wheeled robot evaluation, *IEEE Access*, *8*, 2020, 59618–36.

[17] X.H. Ding, D.D. Zhang, L.G. Zhang, L. Zhang, C.J. Zhang, and B. Xu, Fault detection for automatic guided vehicles based on the two-tower model, *Proc. of the 7th International Conf. on Intelligent Computing and Signal Processing (ICSP)*, Xi'an, 2022, 511–5.

[18] B.Y. Wang, D.Y. Huo, Y.Y. Kang, and J. Sun, AGV status monitoring and fault diagnosis based on CNN, *Journal of Physics: Conference Series*, *2281*(1), 2022, 012019.

[19] X.H. Ding, D.D. Zhang, L.G. Zhang, L. Zhang, C.J. Zhang, and B. Xu, Fault detection for automatic guided vehicles based on decision tree and LSTM, *Proc. of the 5th International Conf. on System Reliability and Safety (ICSRS)*, Palermo, 2021, 42–6.

[20] Z. Miao, F. Zhou, X. Yuan, Y. Xia, and K. Chen, Multi-heterogeneous sensor data fusion method via convolutional neural network for fault diagnosis of wheeled mobile robot, *Applied Soft Computing*, *129*, 2022, 109554.

[21] Z. Miao, Y. Xia, F. Zhou, and X. Yuan, Fault diagnosis of wheeled robot based on prior knowledge and spatial-temporal difference graph convolutional network, *IEEE Transactions on Industrial Informatics*, *19*(5), 2023, 7055–65.

[22] Ö. Gültekin, E. Cinar, K. Özkan, and A. Yazıcı, Multisensory data fusion-based deep learning approach for fault diagnosis of an industrial autonomous transfer vehicle, *Expert Systems with Applications*, *200*, 2022, 117055.

[23] L. Van Der Maaten and G. Hinton, *Visualizing data using t-SNE*, *Journal of Machine Learning Research 9*(11), 2008, 2579–2605.

[24] S. Arora, W. Hu, and P.K. Kothari, An analysis of the t-SNE algorithm for data visualization, *Proc. of the Conf. on Learning Theory*, Stockholm, 2018, 1455–62.

[25] *Deep learning.* (Cambridge, MA: MIT Press, 2016).

[26] W. Zhang, G.L. Peng, C.H. Li, Y.H. Chen, and Z.J. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, *Sensors*, *17*(2), 2017, 425.

[27] D.D. Peng, Z.L. Liu, H. Wang, Y. Qin, and L.M. Jia, A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains, *IEEE Access*, *7*, 2019, 10278–93.

[28] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, *Physica D: Nonlinear Phenomena*, *404*, 2020, 132306.

[29] W. Lu, J. Li, J. Wang, and L. Qin, A CNN-BiLSTM-AM method for stock price prediction, *33*, 2021, 4741–53.

[30] R. Dey and F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, *Proc. of the IEEE 60th International Midwest Symp. on Circuits and Systems (MWSCAS)*, Boston, MA, 2017, 1597–600.

[31] Y.Lecun. LeNet-5, *Convolutional Neural Networks, 20*(5), 2015, 14.

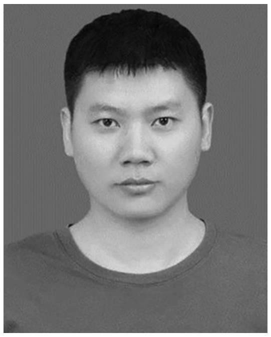
Biographies

Zhi Qiu received the B.S. degree in automation and the M.S. degree in control science and engineering from Southwest Petroleum University, Chengdu, China, in 2003 and 2006, respectively. Qiu is currently an Associate Professor with the School of Mechatronic Engineering, Southwest Petroleum University, Chengdu, China. Her research interests include fault diagnosis of electromechanical systems, robot control, and measurement and control systems for oil and gas equipment.

Zeyu Xu received the B.S. degree in mechanical and electronic engineering from the Changchun University of Science and Technology, Changchun, China, in 2020. Xu is currently pursuing the graduation degree with Southwest Petroleum University. His research interests include deep learning and transfer learning for fault diagnosis.

Min Luo received the B.S. degree in automation from Chongqing University, Chongqing, China, in 2001, the M.S. degree in measurement techniques & measuring instruments from Southwest Petroleum University, Chengdu, China, in 2009, and the Ph.D. degree in control science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2021. Since 2001, Dr.Luo has been a Lecturer with the School of Electrical Engineering and Information, Southwest Petroleum University. Her research interests include robot path planning, trajectory tracking, and intelligent control.

Simon X. Yang (S'97–M'99–SM'08) received the B.Sc. degree in engineering physics from Beijing University, Beijing, China, in 1987, the first M.Sc. degree in biophysics from the Chinese Academy of Sciences, Beijing, in 1990, the second M.Sc. degree in electrical engineering from the University of Houston, Houston, TX, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 1999. He is currently a Professor and the Head of the Advanced Robotics and Intelligent Systems Laboratory, University of Guelph, Guelph, ON, Canada. His research interests include robotics, intelligent systems, sensors and multi-sensor fusion, wireless sensor networks, control systems, intelligent transportation, and computational neuroscience. Prof. Yang he has been very active in professional activities. He serves as the Editor-in-Chief of *International Journal of Robotics and Automation*, and an Associate Editor of IEEE Transactions on Cybernetics, and several other journals. He was the General Chair of the 2011 IEEE International Conference on Logistics and Automation, and the Program Chair of the 2015 IEEE International Conference on Information and Automation. Among many of his awards, he was a recipient of the Distinguished Professor Award at the University of Guelph.



Tao Chen received the B.S. degree in measurement and control technology and instrument from Southwest Petroleum University, Chengdu, China, in 2020. Chen is currently pursuing the graduation degree with Southwest Petroleum University. His research interests include robotics and automation control.