

PAPER • OPEN ACCESS

Intelligent Vehicle Systematic Design Based on Arduino and Raspberry Pi

To cite this article: Yulin Liu *et al* 2023 *J. Phys.: Conf. Ser.* **2632** 012017

View the [article online](#) for updates and enhancements.

You may also like

- [A Modified Mixed Car-following Model Considering that the Connected and Intelligent Vehicle and Non-connected Vehicle](#)

Junjie Zhang, Miaomiao Liu and Zhentian Sun

- [Research on Intelligent Vehicle Trajectory Tracking Control](#)

Boqing Zhu, Feng Du, Zhehui Shi et al.

- [Simulation test of driving assistance system based on virtual scene](#)

Xing Meng, Haiyun Gan and Ying Yan



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Intelligent Vehicle Systematic Design Based on Arduino and Raspberry Pi

Yulin Liu^a, Xiaolu Liu^a, Chunguang Lu^b, Lei Song^b, Guoyu Cui^a, Haifeng Qian^a,
Nick Nianxiong Tan^{*a}

^a Vango Technologies, Inc., Hangzhou, Zhejiang, China

^b Marketing Service Center, State Grid Zhejiang Electric Power Co., Ltd., Hangzhou, Zhejiang, China

*Corresponding author's e-mail: surddar@163.com

Abstract. The intelligent vehicle designed in this paper can realize functions, such as safety detection, visual identification, remote control and manipulator grasping, and so on. Arduino MEGA is used as the main control board to send signal messages to drive vehicles. Wi-Fi module is used to receive messages to remote control vehicles. The ultrasonic and infrared module is used to realize object detection around vehicles. To realize complex route movement, raspberry pie is used for visual recognition and path planning. Data is sent to Arduino for judgment in real time. Finally, it is verified that the design effectively improves the path-planning ability and obstacle-avoidance function in a sample vehicle.

1. INTRODUCTION

With the rapid development of intelligent technology, smart homes have become ubiquitous in people's lives [1]. The emergence of high-tech products, such as intelligent speakers, intelligent cleaners, robotic dogs, intelligent sorting machines, and drones, prioritizes serving people and integrates communication, network, computer, automatic control, and other technologies. As a mobile part of intelligent products, research on smart cars in the home is also becoming increasingly in-depth [2].

Therefore, this article designs an intelligent car based on Arduino and Raspberry Pi control, which can be equipped with various platforms to meet the needs of intelligent services. From a fundamental perspective, it includes sensing, control, and visual components, which can intelligently complete work tasks and wirelessly connect to mobile devices. Intelligent cars achieve different functions through different system programs and modules and can meet the needs of general household use.

2. OVERALL SCHEME DESIGN

We design and research an intelligent car with personalized and expandable functions, mainly composed of Arduino and Raspberry Pi as the main control unit and different functional modules to form a multi-functional intelligent car. Among them, the measurement control management module that can be assembled by enterprise users mainly consists of Bluetooth module, Wi-Fi module, infrared receiving module, ultrasonic module, temperature and humidity module, smoke alarm module, cleaning module, robotic arm module, and voice recognition module. These modules form the control chain of the intelligent car. The overall design scheme of the intelligent car hardware system is shown in Figure 1.



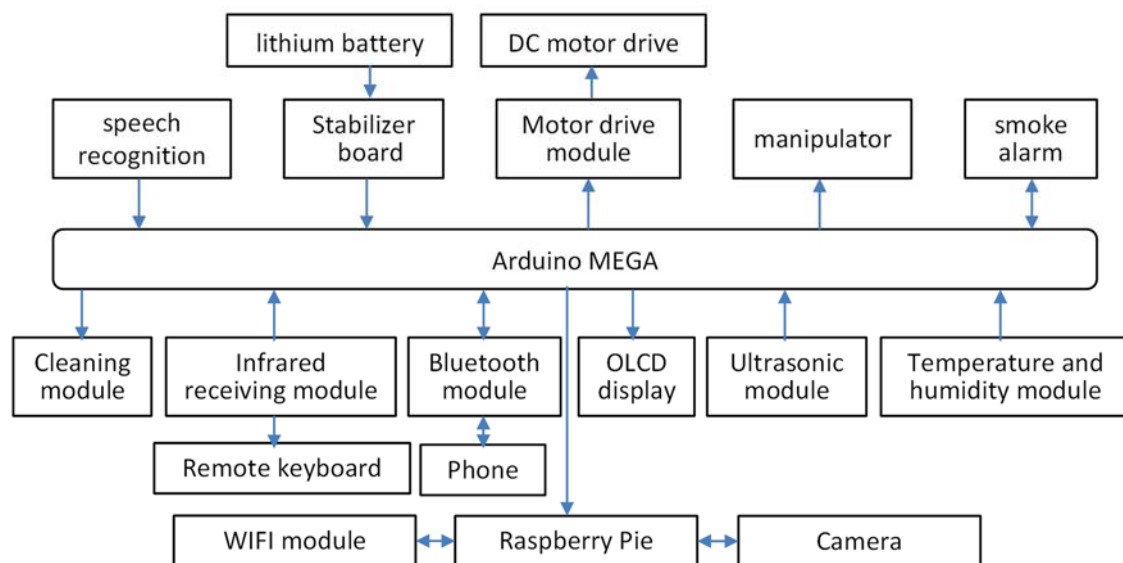


Figure 1. Structure of the Robot Trolley System

2.1. Drive Control Design

The Arduino MEGA main control board controls four motors by outputting data to the motor drive board [3-4]. By using the PWM method to control the motor, it is possible to control both the speed and direction of rotation of the motor [5]. When the enabling end EN is at a high level and the direction end IN1 input pin is at a high level, the motor rotates forward; When the IN2 input pin is high, the motor reverses. The control mode of the left and right motors is the same, and the specific motion logic is shown in Table 1.

Table 1. Intelligent Car Motion Logic Table

Enable A	Enable B	Left motor		Righe motor		Left motor Activity	Right motor activity	Motion
		IN1	IN2	IN3	IN4			
1	1	1	0	1	0	Foreward	Foreward	Toward
1	1	1	0	0	1	Foreward	Reversal	Turn right
1	1	1	0	1	1	Foreward	Stop	Turn right in place with the right motor as the center
1	1	0	1	1	0	Reversal	Foreward	Turn left
1	1	0	1	0	1	Reversal	Reversal	Retreat
1	1	0	1	1	0	Reversal	Foreward	Turn right in place with the left motor as the center

The code transmitted through infrared obstacle avoidance controls the high and low levels of the direction end to control the motor rotation direction and speed, enabling the intelligent car to move freely [6]. Figure 2 is a partial code for the motor drive of the intelligent car.

```

void RUN_Normal(int left_speed, int right_speed, int max_speed)
{
    if(left_speed>0) { digitalWrite(motor_leftfront,HIGH);
                     analogWrite(motor_leftfront_pwm,255-map(255-left_speed,0,255,max_speed,0));
                     digitalWrite(motor_leftrear,HIGH);
                     analogWrite(motor_leftrear_pwm,255-map(255-left_speed,0,255,max_speed,0)); }
    else if(left_speed<0) { digitalWrite(motor_leftfront,LOW);
                     analogWrite(motor_leftfront_pwm,map(abs(left_speed),0,255,0,max_speed));
                     digitalWrite(motor_leftrear,LOW);
                     analogWrite(motor_leftrear_pwm,map(abs(left_speed),0,255,0,max_speed)); }
    else if(left_speed==0) { digitalWrite(motor_leftfront,LOW);
                     analogWrite(motor_leftfront_pwm,0);
                     digitalWrite(motor_leftrear,LOW);
                     analogWrite(motor_leftrear_pwm,0); }
    if(right_speed>0) { digitalWrite(motor_rightfront,HIGH);
                     analogWrite(motor_rightfront_pwm,255-map(255-right_speed,0,255,max_speed,0));
                     digitalWrite(motor_rightrear,HIGH);
                     analogWrite(motor_rightrear_pwm,255-map(255-right_speed,0,255,max_speed,0)); }
    else if(right_speed<0) { digitalWrite(motor_rightfront,LOW);
                     analogWrite(motor_rightfront_pwm,map(abs(right_speed),0,255,0,max_speed));
                     digitalWrite(motor_rightrear,LOW);
                     analogWrite(motor_rightrear_pwm,map(abs(right_speed),0,255,0,max_speed)); }
    else if(right_speed==0) { digitalWrite(motor_rightfront,LOW); analogWrite(motor_rightfront_pwm,0);
                     digitalWrite(motor_rightrear,LOW); analogWrite(motor_rightrear_pwm,0); }
}

```

Figure 2. Example of the motor drive code

2.2. Visual Recognition Module

The visual recognition module mainly uses Raspberry Pie for image recognition and transmits the recognition data to Arduino [7-8]. To achieve visual distance measurement, it is necessary to quickly obtain the image in front of the actual coordinate point. In order to speed up image processing, data processing schemes based on SVM (Support Vector Machine) or VFW (Window Operating System) are not used. Instead, OpenCV's library function is used to compare data processing. By taking photos with a front-facing camera, the image is preprocessed and image features are extracted. To complete the information recognition or data acquisition of digital images, there are five steps: image information acquisition, image processing, obtaining image digital features, classifier image training, and automatic recognition. In order to achieve the accuracy of automatic recognition, there are generally two solutions in program design: one is to use the function CvGrabFrame to quickly capture features consistent with database images from the external environment, and the other is to use the function CvRetrieveFrame to match the images captured by the camera with the database [9]. The second solution is to compare the processed images captured by the camera with the data in the raspberry pie memory, which results in a much higher matching rate than the original image. Therefore, the second solution is adopted to achieve visual recognition of the intelligent car, as well as subsequent functions such as target locking and grasping.

3. IMAGE PROCESSING ALGORITHMS

The CP-ACGAN (classification algorithm) algorithm for image recognition is used in image processing. The traditional CGAN (condition generation countermeasure network) algorithm generates unsupervised countermeasure networks, while the CP-ACGAN algorithm applies the Generative adversarial network to the supervised learning method to make the database features correspond to the generated image features [10]. This algorithm has the characteristics of fast recognition speed and high accuracy performance. The following is the algorithm structure, as shown in Figure 3.

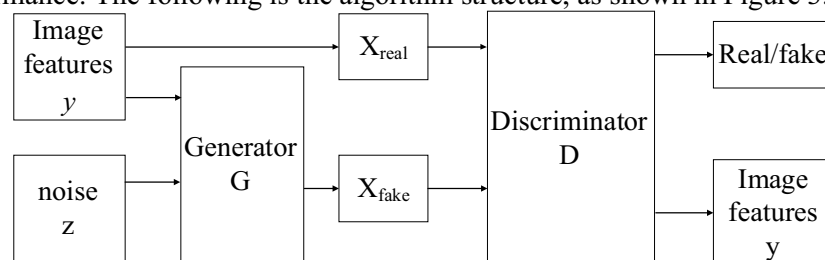


Figure 3. CP-ACGAN algorithm structure

From the algorithm structure (Figure 3), it can be seen that in this network structure, when the sample is output in the discriminator D and generator G interfaces, there is no discriminant term for true or false feature data. At the same time, in order to ensure effective matching of the sample in discriminator D, feature point matching was reintroduced by utilizing the features of the tested object for feature matching. In order to effectively reduce the Loss function value of generator G and discriminator D, the sample read is regarded as supervision data with image characteristics, the processed result data is set as virtual data with image characteristics, and then the Softmax classifier is connected in the judgment network layer, so that the Loss function of supervision data can be obtained as:

$$\begin{aligned}
 L_{\text{supervised}} &= -E(x, y) \sim p_{\text{data}}[\ln p(y'|x, y' < K + 1)] \\
 &= -\frac{1}{N} \ln p(y'|x, y' < K + 1) \\
 &= -\frac{1}{N} \ln \frac{\exp(< y, y' >)}{\sum_{i=1}^K \exp y'_i} \\
 &= -\frac{1}{N} (< y, y' > - \ln(\sum_{i=1}^K \exp y'_i)) = CE(y, y') \quad (1)
 \end{aligned}$$

where N is the number of samples in training, E is the predicted expected value, x is the input sample, $p(x)$ is the distribution function of fitting the real sample, y is the sample image feature, is the predicted image feature of the sample, and $<\cdot>$ is the inner product. Thus, the Loss function of the actual sample data is:

$$L_{\text{real}} = L_{\text{supervised}}.$$

For the calculated data, there are two sources of error: firstly, for the $K+1$ class sample generated by discriminator D, the false sample has a certain probability loss. The other part is the difference loss between the input image feature y of generator G and the output image y'_{fake} feature of discriminator D.

Here, we let $L_{\text{unsupervised}}$ be the probability expected loss of the false sample class, and $y'_{K+1} = 0$. It can be obtained:

$$\begin{aligned}
 L_{\text{unsupervised}} &= -E_{x \sim p_g}[\ln p(y'=K+1|x)] \\
 &= -\frac{1}{N} \ln p(y' = K + 1|x) \\
 &= -\frac{1}{N} \ln \frac{\exp y'_{K+1}}{\sum_{i=1}^{K+1} \exp y'_i} \\
 &= \frac{1}{N} \ln(1 + \sum_{i=1}^K \exp y'_i). \quad (2)
 \end{aligned}$$

In order to ensure that the input image features of each generated sample in the network are consistent with the real sample, the difference loss value between the generated sample characteristics y'_{fake} and the input feature y is $CE(y, y'_{\text{fake}})$. It is known that the function sample loss is:

$$L_{\text{fake}} = 0.5 \times (L_{\text{unsupervised}} + CE(y, y'_{\text{fake}})). \quad (3)$$

During the recognition process, the generator and discriminator parameters need to be updated, so it is necessary to construct the errors of the generator and discriminator separately.

The Loss function of generator D is:

$$L_D = 0.5 \times (L_{\text{real}} + L_{\text{fake}}). \quad (4)$$

The Loss function of generator G is:

$$L_G = 0.5 \times (L_{FM} + L_{\text{unsupervised}}). \quad (5)$$

Among them, $L_{FM} = ||E_{x-Pdata}[f(x)] - E_{z \sim p_z}[f(G(z))]|^2_2$, representing the feature point matching of the binomial loss value.

4. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the effectiveness and reliability of the CP-ACGAN algorithm. We design 10000 training samples and 2000 test samples, using two-dimensional black and white images with 28 pixels each $\times$ 28 pixels. The experiment involves training 50 samples, with one sample per training session. Both generator G and discriminator D are optimized by Adam, and the Learning rate is 0.0002. The results of four algorithms are compared in the visual recognition experiment. The average pooling CNN (Convolutional neural network) algorithm, the maximum pooling CNN algorithm, the ACGAN algorithm and the CP-ACGAN algorithm are used respectively. Table 2 shows the mean and variance of accuracy for the four visual algorithms.

Table 2. Comparison of Mean and Variance of Four Algorithms

Algorithm	Mean Value	Variance
Average Pooling CNN Algorithm	0.994980	6.000 000e-09
Maximum Pooling CNN Algorithm	0.994144	3.264 000e-09
ACGAN Algorithm	0.994116	4.069 440e-07
CP-ACGAN Algorithm	0.995604	1.923 840e-09

From Table 2, the accuracy error of the CP-ACGAN algorithm is relatively small, which is compared with the other three algorithms. The variance of the CP-ACGAN algorithm is only 1923e-09, which is much smaller than the other three algorithms. Therefore, it has good accuracy. Meanwhile, the accuracy of the CP-ACGAN algorithm is 99.5604%, which is also higher than the other three algorithms. At the same time, the CP-ACGAN algorithm also has high operational stability, making visual recognition more stable and accurate.

From Table 2, it can be seen that the CP-ACGAN algorithm can achieve good image quality during image processing. The processed image has better clarity and boundary effect than the ACGAN algorithm because the algorithm improves the convolution method. Therefore, the CP-ACGAN algorithm has a higher recognition rate, better stability, and faster recognition speed. Applying the CP-ACGAN algorithm to the visual recognition of smart cars can achieve accurate recognition of the mechanical grippers of the cars.

5. CONCLUSION

This design is to achieve intelligent recognition, obstacle avoidance, grasping, and cleaning functions of the intelligent car by combining various modules reasonably. A visual recognition function based on the CP-ACGAN algorithm was designed for an intelligent car based on Arduino and Raspberry Pi. The input and output data structure of the feature discriminator can be completely changed by performing feature recognition matching to reduce the Loss function and improve the recognition rate. This recognition method has good classification performance, high stability, and good scalability when applied to intelligent vehicles. The intelligent car designed in this article can meet the general needs of people's lives, improve home intelligence, and have diverse operations, expandable functions, and good stability performance.

ACKNOWLEDGMENT

This paper is funded by the project (5700-202341281A-1-1-ZN) of the State Grid Corporation of China in 2023.

REFERENCES

- [1] Wu Botao, Kong Jinping, Wang Xiang Design and Implementation of an Intelligent Car Based on Arduino and Raspberry Pi [J] *Electronic Design Engineering*, 2017, 025 (15): 58-61

- [2] Zhong Qiyuan, Xu Jinbiao Exploring the Future of Smart Home [J] *Electronic World*, 2016 (3): 67-69
- [3] Wang Senlong, Wang Zhe. Hardware Design of an Intelligent Track Avoidance Vehicle Based on the Arduino Platform [J]. *Electronic World*, 2018 (9): 2. DOI: CNKI: SUN: ELEW.0.2018-09-089
- [4] Wang Senlong, Wang Zhe. Design of Automatic Obstacle Avoidance System for Intelligent Cars Based on Arduino [J]. *Chinese Information*, 2018, 000 (009): 228-229. DOI: 10.3969/j.issn.1003-9082.2018.09.203
- [5] Zhang Jie, He Wentao, Feng Huaxing PWM Circuit Design Based on DC Brushless Motor [J] *Microelectronics and Computers*, 2021, 38 (3): 84-88
- [6] Wang Nan, Zhao Ji, Zhao Biao, etc Intelligent classification trash bin based on raspberry pie [J] *Science and Technology Innovation*, 2021 (12): 167-168
- [7] Zhou Linyong, Xie Xiaoyao, Liu Zhijie, et al Image recognition algorithm based on ACGAN [J] *Computer engineering*, 2019, 45 (10): 246-252, 259
- [8] Zhu Kefan, Wang Jiegui, Liu Youjun. Radar target recognition algorithm based on data augmentation and WACGAN under small sample conditions [J]. *Journal of Electronic Science*, 2020, 48 (6): 8. DOI: 10.3969/j.issn.0372-2112.2020.06.012
- [9] Wei Lianchun. Research on Image Recognition Based on Deep Network ACGAN [J]. *Computer enthusiasts*, 2019, 01 (01): 131. DOI: CNKI: SUN: DNMI.0.2019-01-113
- [10] Wu Botao, Kong Jinping, Wang Xiang. Design and Implementation of Intelligent Cars Based on Arduino and Raspberry Pi [J]. *Electronic Design Engineering*, 2017 (15): 4. DOI: 10.3969/j.issn.1674-6236.2017.15.015