



Chunxiang Yan <sup>1,2</sup> and Xiying Li <sup>1,\*</sup>

- School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 510006, China; yanchx23@mail2.sysu.edu.cn
- <sup>2</sup> Guangzhou Automobile Group Co., Ltd., Automotive Engineering Research Institute, Guangzhou 511434, China
- \* Correspondence: stslxy@mail.sysu.edu.cn

Abstract: In complex terrain, such as uneven roads or irregular terrain, two-wheeled heavy-duty selfbalancing vehicles are easily affected by external interference factors, causing rollover or rendering the vehicle unable to move, which poses a greater challenge to its stability control. Therefore, it is necessary to establish a kinematic model of a two-wheeled vehicle and design a control system to study its driving stability. This paper aims to study the stability control system of a two-wheeled self-balancing vehicle under complex terrain. First, a self-balancing vehicle modeling method based on complex terrain is designed. By analyzing the motion characteristics of the self-balancing vehicle, a kinematic model suitable for complex terrain is established, which provides a basis for subsequent control algorithms. Secondly, a precise control system is designed for different terrain conditions, and parameters such as vehicle attitude, speed and acceleration are adjusted through the Proportional-Integral–Derivative (PID) control algorithm to achieve the smooth operation of the self-balancing vehicle in complex terrain. In addition, a vehicle-mounted camera is used to capture terrain images in real time, and different terrains are accurately identified through the terrain recognition algorithm based on deep learning, thereby determining the friction coefficient and effectively improving the stability of the self-balancing vehicle on complex terrain. The experimental results show that the designed control system can enable the self-balancing two-wheeled vehicle to achieve stable balance control in different terrains, and has good applicability and stability.

**Keywords:** two-wheeled self-balancing vehicle; motion control; PID control; deep learning; terrain recognition

## 1. Introduction

Nowadays, the complexity and diversity of the working environment have placed increasing demands on the mechanical structure design of mobile robots. Mobile robots are increasingly used to replace humans in harsh environments to complete some tasks, such as earthquake rescue, fire detection, the transportation of dangerous goods, etc. Considering that robots need to respond quickly and encounter complex road conditions when working, among the existing technologies, wheel-legged robots that combine the motion advantages of wheeled and legged mobile robot mechanisms have received widespread attention [1,2]. A wheel-legged robot is a nonlinear, under-actuated, strongly coupled multi-variable system. The problem of motion balance has always been a key issue in the research of wheel-legged robots. The study of its motion control has great theoretical and practical significance.

With the increasing popularity of mobile robots in recent years, the structure of robots has become more and more novel, such as the foreign Boston Dynamics bipedal humanoid robot and the two-wheeled robot of the Zurich team in Switzerland. In recent years, Tencent Labs has released the latest wheel-legged robots in China and the Xingtian robot from



Citation: Yan, C.; Li, X. Research on Stability Control System of Two-Wheel Heavy-Load Self-Balancing Vehicles in Complex Terrain. *Appl. Sci.* **2024**, *14*, 7682. https://doi.org/10.3390/ app14177682

Academic Editor: Suchao Xie

Received: 12 July 2024 Revised: 19 August 2024 Accepted: 20 August 2024 Published: 30 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Benmo Technology Company (Dongguan, China). Their own topological structures have their own merits, and the current research on this type of robot is still at a critical stage. Among them, there are many researchers of humanoid robots [3,4]. There are two very important criteria for the quality of humanoid robots. The first criterion is controllability, and the second criterion is maneuverability. When humanoid robots solve their mobility problems, they mainly use two methods: bipedal walking and wheeled. Bipedal walking is a method often used by humanoid robots. Many research institutions have adopted corresponding methods to realize the bipedal walking of robots. Both feet have superior performance on rough roads, going up and down stairs, and in complex outdoor environments. However, its maneuverability is slightly insufficient, and its traveling speed on flat roads cannot meet the ideal requirements. Moreover, the mechanical structure is complex, making it difficult to manufacture and design.

Wheeled robots are also a solution for humanoid robot walking. The high mobility of wheeled vehicles on level ground and their flexibility in small spaces are what motivate researchers to study them. Wheeled robots require at least three wheels to achieve static stability, while common vehicles are generally equipped with four wheels to ensure stability at high speeds. However, in the case of non-flat roads, four-wheel vehicles need to design corresponding suspension systems to solve the over-constraint problem caused by the four wheels, resulting in an increase in the structural complexity of the system. The two-wheeled self-balancing robot has the characteristics of simple structure and flexible movement. Its left and right wheels are connected on the same axis. Two-wheeled self-balancing robots can complete complex movements and operations in a small space. Such superior performance is not available in multi-wheeled robots [5].

With the rapid development of computer vision, its applications in various fields are becoming more and more widespread. Terrain recognition, as one of the important research contents in the current field of robotics, has also made significant progress with the help of deep learning. The goal of terrain recognition is to achieve the classification of terrain types, the perception of ground conditions, and path planning through automatic detection and identification of surface features around the robot, which has important application value in the field of intelligent robots [6–8]. Terrain recognition technology is particularly important for robot applications, especially outdoor robots. It is mainly based on advanced convolutional neural network (CNN) frameworks, such as VGGNet [9], ResNet [10] and DenseNet [11], etc., which can automatically learn feature representations from terrain images to obtain excellent performance [12]. Fei et al. [13] proposed a deep coding pool network based on ResNet to identify flat, obstacle-laden and complex terrain, and assist the robot in gait switching during movement. In addition, for mobile deployment, some lightweight networks have been recently proposed, such as ShuffleNet [14], GhostNet [15] and MobileNet [16]. These networks require fewer parameters and calculations, but are more accurate and convergent. The speed is not as good as the above models with larger parameters.

In this work, we designed a stability control system for a two-wheeled heavy-duty self-balancing vehicle to achieve the stable operation of the self-balancing vehicle on complex terrain. First, its kinematic characteristics are analyzed and modeled to understand the vehicle's dynamic properties and control requirements. Secondly, a real-time terrain recognition method was established based on deep learning technology to accurately identify the terrain of the self-balancing vehicle and determine the friction coefficient, thereby ensuring the stability of the self-balancing vehicle and achieving precise control. Finally, the effectiveness and stability of the proposed control system in complex terrain were verified through experiments. Figure 1 is a structural diagram of the stability control system of a two-wheeled self-balancing vehicle. The main contributions of this paper are summarized below.

1. Design a heavy-duty two-wheeled self-balancing vehicle modeling method to make the center of mass calibration more accurate.

- 2. Determine the friction coefficient through terrain recognition results to ensure the stability of the self-balancing vehicle and achieve precise control.
- 3. Propose a lightweight terrain recognition method based on deep learning, introduce a coordinate attention mechanism to improve the network's feature extraction capabilities for different types of terrain, and construct an auxiliary loss function to optimize the network.

The remainder of this paper is organized as follows. Section 2 introduces related work, including wheel-legged robots, terrain recognition, and self-balancing control strategies. Section 3 introduces the research methods, including establishing a dynamic model of a two-wheeled self-balancing vehicle, a self-balancing control algorithm and a lightweight terrain recognition network. Section 4 introduces the terrain recognition results and experimental results. Section 5 summarizes the work of this paper.



**Figure 1.** Two-wheeled self-balancing vehicle stability control system structure diagram, including modeling, terrain recognition, and stability control.

### 2. Related Work

Next, we will introduce the development history of the types of wheel-legged robots, the research on terrain recognition methods, and the self-balancing control strategy.

## 2.1. Wheel-Legged Balancing Robot

Robots traveling in complex outdoor environments need to have strong obstacle avoidance performance, climbing performance, chassis stability and flexibility. There are four main types of existing robot walking mechanisms: four-wheeled, bipedal, four-legged and crawler. In recent research, wheel-legged robots have received widespread attention as a multi-modal motion mechanism. The robot combines the characteristics of wheels and legs and can travel in wheel mode on flat ground and switch to leg mode on complex terrain to overcome obstacles and irregular terrain. Wheel-legged robots are a combination of legged robots and wheeled robots. They have humanoid or bionic leg structures and the characteristics of wheeled robots. Two-wheeled and wheel-legged bionic robots not only have the advantages of two-wheeled self-balancing robots, but also take into account the flexible humanoid or bionic characteristics of legged robots, and have huge potential application prospects in various fields. Figure 2 shows some existing pictures of wheel-legged robot from

Tencent Robotics X Lab, (d) is from ETH Zurich, (e) is a balanced infantry robot made by Harbin Engineering University, and (f) is the Xingtian robot from China Benmo Technology Company.

Research on wheel-legged robots aims to explore their applicability and advantages in different environments. A key research direction consists of mode switching and control strategies for robots. In order to achieve reliable mode switching and smooth motion, researchers have proposed various mode switching strategies [17,18]. In addition, the perception capabilities and environment modeling of wheel-legged robots are also a key aspect of the research. In order to achieve autonomous navigation and obstacle avoidance, robots need to accurately sense and understand the surrounding environment. Therefore, researchers use sensors, cameras, and inertial measurement units to obtain information about terrain and obstacles and conduct environment modeling. This information is used for tasks such as terrain recognition, path planning, obstacle avoidance, and environment perception [19,20].



Figure 2. Different types of wheel-legged robots available.

### 2.2. Terrain Recognition

Terrain recognition is crucial for gait planning, speed control and surrounding environment observation of outdoor mobile robots. Before the emergence of CNN models, traditional terrain recognition solutions usually performed classification by extracting basic visual features such as the color and texture of terrain images. Li et al. [21] proposed an extreme learning method with filters and clustering algorithms to classify terrain images, and achieved remarkable results. Ebadi et al. [22] extracted color features from road digital images collected by cameras installed on cars, and then used multi-layer perceptron to classify the color features, thereby achieving the classification of four types of soil, grass, stone, and asphalt. For the purpose of terrain classification, Liu et al. [23] designed a complex terrain sample segmentation scheme, using a combination of graph segmentation and watershed segmentation to identify terrain images.

The traditional vision-based terrain recognition method has many shortcomings. The recognition effect is not good under the influence of complex terrain environment, lighting and other external environments, and it cannot meet the real-time needs. With the continuous development of CNN, deep learning methods have gradually become mainstream, which can effectively solve the limitations of traditional terrain recognition methods and improve recognition accuracy and reliability [24]. Liu et al. [25] first introduced the deep learning method into scene terrain recognition, and proposed a terrain classification method based on a deep sparse filtering network, which combines the spatial information between image pixels with the input data, and uses the deep learning network to automatically learn features from the input data to classify the terrain. Wei et al. [26] proposed a lightweight and efficient deep neural network for pixel-level terrain recognition in complex environments to achieve the global perception of outdoor environments. The deep learning network can not only extract the spatial geometric features of terrain images, but also extract the color texture details of terrain images, achieving end-to-end learning [27–29].

# 2.3. Self-Balancing Control Strategy

The wheel-legged robot is a robot system with the ability to move autonomously. It has a unique wheel-foot structure and can achieve stable movement on uneven ground. In order to maintain the balance of the robot, self-balancing control strategy is one of the key research directions. In related research, a variety of self-balancing control strategies

have been proposed to improve the stability and motion performance of wheeled robots. A common self-balancing control strategy is the PID controller-based approach. The PID controller calculates and controls the robot's output by measuring the error between the robot's current state and the target state to maintain the robot near its equilibrium position. This method achieves self-balancing control by continuously adjusting the speed and attitude of the robot. Tran et al. [30] proposed a fuzzy LQR PID control for a bipedal wheeled balancing robot to maintain stability under uncertainty and variable height. LQR control is used to stabilize the robot and control its movement, and PID control is used to control the robot's posture and help maintain balance. Zhang et al. [5] used PID control strategy to perform balance control and speed control on a two-wheeled wheel-legged robot. They placed the balance controller in the inner loop of the speed controller to ensure the priority of balance control, by changing the target inclination angle of the balance controller to control the speed. Liu et al. [1] proposed a variable-height dynamic balance control strategy based on a PID controller, using a PID controller for dynamic balance control. By constraining the center of mass to an axis perpendicular to the ground, at the center of the two wheels, the height can be changed while maintaining dynamic balance.

## 3. Methods

This section first introduces the structure and kinematics of the two-wheeled selfbalancing vehicle. By accurately establishing the kinematic model between each joint angle and joint position coordinates, it provides the basis for subsequent control algorithms. A lightweight terrain recognition algorithm based on attention mechanism and auxiliary loss function is proposed to accurately identify different terrains and thereby adaptively determine the friction coefficient. Finally, the PID control algorithm is used to adjust vehicle attitude, speed, acceleration and other parameters to achieve the smooth operation of the self-balancing vehicle in complex terrain. The overall flow chart of the control system is shown in Figure 3.



Figure 3. Wheel-legged robot stability control flow chart.

#### 3.1. Establishment and Analysis of Kinematic Models

Since the wheel-legged robot is equipped with drive motors on its hip and knee joints, and it is assumed that the left and right joints move synchronously, the robot can be abstracted into a wheeled inverted pendulum model with a variable structure [31]. The forward kinematics modeling method is used to derive the posture of each link relative to the base frame, and then the position of the robot's center of mass is calculated. In order to derive the position and tilt angle of the robot's center of mass, the general formula for calculating the center of mass is first given as follows:

$$X_{full}(p_x, p_y, p_z) = \frac{\sum m_i X_i^{iw}(q)}{\sum m_i}$$
(1)

$$X_i^w(q) = T_i^w(q) X_i^i \tag{2}$$

The center of mass positions px, py, pz are vectors about the current posture of the robot,  $X_{full}$  represents the center of mass position of the robot in the current posture, q is a vector about the joint angle, and mi represents the mass of the *i*-th link.  $X_i^w(q)$  represents

the center of mass position of the *i*-th link in the current posture,  $T_i^w(q)$  is the transformation matrix of the local coordinate system of link *i* relative to the base coordinate system, and  $X_i^i$  is the position of the center of mass of link *i* in the local coordinate system.

The complete mechanical structure and coordinate diagram of the robot are shown in Figure 4. In order to obtain the transformation matrix of the robot, we establish a reference coordinate system. According to the structural parameters of the robot link, the Denavit–Hartenberg (D-H) parameter table of the robot can be listed [32], as shown in Table 1. Among them, *i*1 represents a joint,  $a_{i-1}$  is the angle difference between joint *i*1 and joint *i* on the *z*-axis, and  $l_{i-1}$  is the shortest distance between joint *i*1 and joint *i* on the *z*-axis.  $d_i$  is the shortest distance between  $a_{i-1}$  and  $a_i$ , and  $\theta_i$  is the angle difference between  $a_{i-1}$  and  $a_i$ .  $L_1$  is the length of the calf,  $L_2$  is the length of the thigh,  $L_3$  is the length from the end of the thigh to the carrying platform, *a* is the length of the two wheel axes, *b* is the shoulder width, and *c* is the waist width.

i	$a_{i-1}$	$l_{i-1}$	$d_i$	$ heta_i$
1	0	0	$\frac{a}{2}$	$ heta_1$
2	0	$L_1$	ō	$\theta_2$
3	0	$L_2$	$-\frac{b}{2}$	$\theta_3$
4	0	$L_3$	$-\frac{\overline{c}}{2}$	$-90^{\circ}$

Table 1. D-H parameter table of the two-wheeled self-balancing vehicle.



**Figure 4.** Coordinate diagram of wheel-legged robot, the arrows represent the base coordinate system, and 0–7 represent different joints.

The individual transformation matrix of each link is obtained from the transformation matrix formula of adjacent links, and then the transformation matrix of each link relative to (0) is obtained. Among them, the transformation matrix of the carrying platform (4) relative to the reference coordinate system (0) is:

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \begin{bmatrix} s_{123} & c_{123} & 0 & L_1 c_1 + L_2 c_{12} + L_3 c_{123} \\ -c_{123} & s_{123} & 0 & L_1 s_1 + L_2 s_{12} + L_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(3)

Next, we measure the mass of each link and the position of its center of mass relative to its local frame. Substituting the measurement results into Equations (1) and (2), the center of mass position of the robot relative to the base frame can be solved.

$$X_{full}(p_x, p_y, p_z) = \frac{X_A^0 m_A + X_B^0 m_B + X_C^0 m_C + X_D^0 m_D + X_E^0 m_E}{m_A + m_B + m_C + m_D + m_E}$$
(4)

It can be seen from the above formula that the center of mass positions ( $p_x$ ,  $p_y$ , and  $p_z$ ) are vectors related to the current pose q of the robot, and the tilt angle of the robot can be obtained as:

$$\emptyset = \arctan\left(\frac{p_x}{p_y}\right) \tag{5}$$

When the robot is in dynamic equilibrium,  $\emptyset = 0$ . In order to maintain the overall balance, the waist is also in a horizontal posture, and at the same time, it is also expected to control its overall height, therefore:

$$k_1s_1 + k_2c_1 + k_3c_3 + k_4s_3 + k_4s_3 + k_5 = 0 ag{6}$$

$$\theta_1 + \theta_2 + \theta_3 = \frac{\pi}{2} = 90^\circ \tag{7}$$

$$L_1 s_1 + L_2 s_{12} + L_3 s_{123} = h \tag{8}$$

# 3.2. Research on Control Strategy of Self-Balancing Two-Wheeled Vehicle

This study uses the PID algorithm as the core of the robot's motion balance controller to design a control method for wheel-legged balance robots to achieve stable operation and unilateral obstacle crossing on complex terrain. The mathematical principle of the PID algorithm is as follows:

$$u_k = K_p \cdot e(k) + K_i \cdot \sum_{j=0}^k e_j + K_d \cdot (e_k - e_{k-1})$$
(9)

In the formula,  $u_k$  is the control quantity,  $e_k$  is the current error of the control quantity,  $K_p$  is the proportional coefficient used to adjust the system's response speed to the control quantity, and  $K_p$  is the integral coefficient used to eliminate the error of the control quantity in the steady state of the system.  $K_p$  is a differential coefficient used to suppress system oscillation caused by the control variable during the control process. When the wheel-legged robot moves forward or backward, if the robot load is unknown, there will be a certain error between the calculated center of mass position of the robot and the actual position. If there is only a PD balance controller, the system will lose control due to model errors. A PI speed controller needs to be added to calculate a new balance target inclination angle to eliminate the impact of the error and improve system stability, as shown in Figure 5a. Therefore, the control algorithm consists of a negative feedback balanced upright loop PD controller and a positive feedback speed loop PI controller. Since upright balance is the ultimate control goal, the output of speed control is used as the input of upright control, and the relationship between the two is as follows:

$$\begin{cases} u = K_p \cdot (\varphi - u_1) + K_d \cdot \ddot{\varphi} \\ u_1 = K_{p_1} \cdot e(k) + K_{i_1} \cdot \sum_{j=1}^k e(j) \end{cases}$$
(10)

$$u = K_p \cdot \varphi + K_d \cdot \ddot{\varphi} - K_p \cdot \left[ K_{p_1} \cdot e(k) + K_{i_1} \cdot \sum_{j=1}^k e(j) \right]$$
(11)

where,  $K_p = 17.7$ ,  $K_d = 200.0$ ,  $K_{p_1} = 0.039$ ,  $K_{i_1} = 0.000195$ . When a wheel-legged robot performs a unilateral obstacle course, its knee joint is raised when it encounters an obstacle in the roll angle direction. In order to achieve balance between the left and right calves, the roll angle integral angle of the gyroscope is selected as the deviation, and the roll angular velocity is used as the differential term for PD control, as shown in Figure 5b.

$$u = K_{p_3} \cdot (\varphi - u) + K_{d_3} \cdot \dot{\varphi} \tag{12}$$

where,  $K_{p_3} = -4.5$ ,  $K_{d_3} = -20$ .



#### (a) Forward and backward;

(b) Unilateral obstacle crossing.

**Figure 5.** Unilateral obstacle crossing. (a) Forward and backward, (b) unilateral obstacle crossing, where  $\alpha$  is the speed control signal,  $\beta$  is the balance control signal,  $\nu$  is the actual left and right wheel speed,  $\varphi$  and  $\omega$  are the actual pitch angle and angular velocity,  $\beta'$  is the unilateral balance control signal, and  $\omega'$  is the actual roll angle.

# 3.3. Terrain Recognition and Stability Analysis

In this section, we will focus on the terrain recognition network LA-MobileNet (The "LA" stands for lightweight and high accuracy) and its components, including the coordinate attention mechanism and auxiliary loss function. In addition, we also conducted a detailed analysis of the stability of the two-wheeled self-balancing vehicle on flat and sloped terrain.

### 3.3.1. LA-MobileNet Network

The overall structure of the LA-MobileNet is shown in Figure 6. First, cascaded convolutional layers are used for feature extraction, then its output is input to the global pooling layer to obtain the spatial geometric features of the image, and finally the predicted category is obtained through the classifier. The CA [33] is introduced into the Bneck structure, and global average pooling is performed from the horizontal and vertical spatial directions to obtain direction-awareness and position-awareness information. An auxiliary loss function is used in the middle layer of the network to alleviate the problem of vanishing gradients and improve the generalization ability of the neural network.

The SE [34] attention mechanism in the MobileNetV3 model mainly focuses on internal channel information without considering the impact of position information. In contrast, the CA embeds position information into channel attention, which not only avoids introducing excessive calculations, but also enables the model to obtain richer information. The structure is shown in Figure 6. The CA module can avoid the loss of position information caused by global pooling–2D operations, focus on the width and height dimensions, respectively, and effectively utilize the spatial coordinate information of the input feature map. The output of the c-th channel with height h and width w is shown in Equations (13) and (14). Among them, in the horizontal pooling operation, each row of the attention matrix is summed to obtain a horizontal summary attention vector, as shown in Equation (13). In the vertical pooling operation, each column of the attention matrix is summed to obtain a vertical summary attention vector, as shown in Equation (14).

$$Z_{c}^{h}(h) = \frac{1}{W} \sum_{0 \le i \le W} x_{c}(h, i)$$
(13)

$$Z_{c}^{w}(w) = \frac{1}{H} \sum_{0 \le j \le H} x_{c}(j, w)$$
(14)

where  $x_c(h, i)$  represents the input of the c-th channel in the horizontal direction;  $x_c(j, w)$  represents the input of the c-th channel in the vertical direction; H and W represent the height and width of the input feature map, respectively.



**Figure 6.** (a) Overall structure of LA-MobileNet; (b) Bneck structure with CA; (c) the coordinate attention mechanism structure, where  $f_1$  and  $f_2$  are direction-aware features, and  $f_3$  and  $f_4$  are attention weights in the vertical and horizontal directions.

Coordinate attention first splices the feature maps obtained in the previous stage in the spatial dimension, uses  $1 \times 1$  convolution to compress the channels, and uses the ReLu function to perform nonlinear activation of spatial information in the vertical and horizontal directions. The feature map is then divided into horizontal tensors and vertical tensors. Then,  $1 \times 1$  convolution is used to increase the channel dimension so that the number of channels is consistent with the input feature map. Finally, the Sigmod function is used for nonlinear activation and weighted fusion. The final output result is shown in Equation (15).

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j)$$
(15)

Among them,  $y_c(i, j)$  is the output of the *c*-th channel;  $x_c(i, j)$  is the input feature map;  $g_c^h(i)$  is the attention weight in the horizontal direction;  $g_c^w(j)$  is the attention weight in the vertical direction.

The backbone network we use is MobileNetV3-large, which has a deep network structure and is prone to the problem of vanishing gradients. Inspired by the literature [35], we choose to add an auxiliary classifier on the feature layer of 32 × 32 size. In order to obtain more hierarchical features, a Bneck module is added to the classifier for feature extraction, and then a classifier composed of Dropout, ReLU and Linear layers is used for classification. Finally, the output results of both classifiers are calculated using cross entropy loss and combined according to certain weights.

$$Loss = \beta \cdot Loss_1 - (1 - \beta) \cdot \ln(1 - Loss_2)$$
(16)

Loss<sub>1</sub> is the backbone network loss, and Loss<sub>2</sub> is the auxiliary classification loss. Through experiments, we found that the best effect is when the weight coefficient  $\beta$  is 0.8.

Since there are currently few publicly available terrain datasets, the GTOS-mobile [36] dataset is used for network training. The GTOS-mobile dataset covers 31 types of outdoor ground terrain under different weather and lighting conditions. In order to better apply it to the terrain recognition algorithm, we have reorganized it and retained the 8 common terrain categories in the image dataset, namely soil, pebble, sand, cement, grass, asphalt, brick, and wood\_chips. It contains 35,163 training sets and 1713 test sets. Various terrain maps are shown in Figure 7. We named this dataset GTOS-mobile8. During the training process, we use cutmix and mixup methods for data enhancement, and set the training and test image sizes to  $256 \times 256$ .



**Figure 7.** Different ground terrain datasets. (From (**a**)–(**h**): soil, pebble, sand, cement, grass, asphalt, brick, wood\_chips).

#### 3.3.2. Stability Analysis

In the static simulation, it was determined that the robot is made of stainless steel with a density of 7930 kg/m<sup>3</sup>, and the rubber density of the tire is 1200 kg/m<sup>3</sup>. Through the statistics of the 3D digital model with a 1:1 ratio with the real object, it is known that the mass of the truss composed of non-standard stainless steel parts is 35.3 kg, the mass of the 4 wheels is 10 kg, and the total mass of the 2 hip joints, 2 knee joints, and 2 ankle joints is 36.4 kg. The total mass of the wheeled robot is 81.7 kg, and its center of gravity coordinates are (0,0,0). Based on this, the theoretical analysis of the robot's driving stability on flat ground and slopes is carried out. The C++ programming language was used, and the UG10.0 (Unigraphics NX) 3D digital model design software was used to design the robot in 1:1 size. The ANSYS Workbench2021R2 simulation software was used to construct the finite element space of the digital model.

Straight-line driving stability means that the tires do not slip during straight driving, and the condition is that the tire adhesion force is not less than the driving force  $F_t$  generated by the ankle joint motor. The driving force is the resultant force in the horizontal direction from the ground resistance  $F_f$ , air resistance  $F_w$  and acceleration resistance  $F_j$  during tire rolling. As shown in Figure 8a, the relationship is shown in the following formula.

$$F_t = F_f + F_w + F_j \le F_\mu$$

$$fmg + \frac{1}{2}C_D A\rho v^2 + ma \le \mu mg$$
(17)

In the formula, *f* is the rolling damping coefficient, *g* is the acceleration of gravity,  $C_D$  is the air damping coefficient,  $\rho$  is the air density, *A* is the windward area, *v* is the relative speed of the robot and the air, *m* is the mass of the robot, *a* is the acceleration, and  $\mu$  is the adhesion coefficient between the tire and the road. Under normal circumstances  $C_D$ ,  $\rho$ , *g*, *A* and *m* remain unchanged, so the instantaneous acceleration *a* when the robot travels horizontally and straight mainly depends on the adhesion coefficient  $\mu$  and the rolling damping coefficient *f*. That is, the constraint condition for the wheel-footed robot to run straight on flat ground without slipping is:

$$a \le \frac{\mu m g - f m g - 1/2 C_D A \rho v^2}{m} \tag{18}$$

In the above formula,  $C_D$  is 0.3, air density  $\rho$  is 1.29 kg/m<sup>3</sup>, gravity acceleration  $g = 9.80 \text{ m/s}^2$ , robot windward area  $A = 0.45 \text{ m}^2$ , and mass *m* is 81.7 kg. On cement floors, bricks and asphalt roads, the adhesion coefficient  $\mu$  between the rubber tire and the ground should be 0.7, the rolling damping coefficient *f* should be 0.4, and the maximum straight speed is designed according to 10 m/s. Then, the acceleration  $a \le 2.83 \text{ m/s}^2$  during the straight movement of the wheel-footed robot, otherwise slipping may occur. On grass, sand, soil, gravel and wood chips roads, the adhesion coefficient *f* should be 0.5, and the maximum straight speed is designed according to 6 m/s. Then the acceleration  $a \le 0.94 \text{ m/s}^2$  during the straight movement of the wheel-footed robot, otherwise it may slip.

As shown in Figure 8b. The distance between the grounding points  $O_1$  and  $O_2$  of the two wheels of the robot is L. When the robot is located on a slope with a slope angle a, the contact point between the front wheel and the ground is  $O_1$ , the contact point between the rear wheel and the ground is  $O_2$ , and the height of the center of gravity O from the slope is h, and its projection on the ground moves from point  $P_1$  on the level ground to point  $P_2$  on the sloped ground, the distances between point  $P_2$  and points  $O_1$  and  $O_2$  are  $L_2$  and  $L_1$ , respectively. At this time,  $L_2 < L_1$ , so the pressure of the robot on the ground at point  $O_2$  is greater than that at point  $O_1$ .

The condition for a wheel-footed robot to travel on a slope without tipping longitudinally is that the vertical pressure on the ground at the contact point between the wheels with a higher horizontal position on the slope and the ground is greater than zero. Since the slope will affect the position of the robot's center of gravity at point  $P_2$  on the slope, when the slope is large enough, point  $P_2$  coincides with point  $O_2$ . At this time,  $L_1 = 0$ ,  $L_2 = L$ , the pressure of the robot's front wheels on the ground is 0, and the pressure of the rear wheels on the ground is *G*. Since the friction force provided by the ground for the wheels under extreme slope conditions is only used to offset the component of the robot's gravity parallel to the slope, it will cause the robot to tip over on the slope. At this time, the distance from point  $P_1$  to point  $O_2$  is 0.5 *L*, and *h* is the distance from the center of gravity *O* of the robot to the slope surface, as shown in Figure 8b. The expression of the limit tilt angle according to the geometric relationship is as follows:

$$\beta_{o-\lim} = \arctan \frac{0.5L_3}{h} \tag{19}$$

Combined with the wheel-footed robot body structure, the height of the center of gravity *h* is 0.350 m, the distance *L* between the two wheel axes is 0.532 m,  $L_3$  is 0.710 m, the limit tilt angle for the lateral direction is  $\beta_{o-\text{lim}} = 45.41^{\circ}$ .





(a) Flat ground force analysis.

(b) Slope force analysis.

Figure 8. Stability analysis of wheel-legged self-balancing robot on different terrains.

### 4. Experiments and Results

In this section, we conducted terrain recognition experiments and result analysis based on deep learning, as well as stability experiments and analysis of the two-wheeled self-balancing vehicle in straight driving and obstacle crossing on different terrains.

### 4.1. Terrain Recognition Results

To verify the performance of the proposed LA-MobileNet model, we compare it with several state-of-the-art classification methods, including VGG16, ResNet50, ShuffleNetV2, MobileNetV3, EfficientNet [37], InceptionV3 [38] and DenseNet. The experimental results are shown in Table 2. It can be seen that LA-MobileNet achieved better results. Compared with Resnet50, a model with a larger number of parameters, the indicators Accuracy, Recall, F1-score and Precision have increased by 0.7%, 1.35%, 1.34% and 0.87%, respectively. Compared with the lightweight model ShuffleNetV2, the indicators Accuracy, Recall, F1-score and Precision have increased by 2.16%, 4.78%, 4.33% and 2.05%, respectively.

Method	Accuracy	Recall	F1-Score	Precision	
VGG16 [9]	0.8844	0.8036	0.7834	0.7896	
Resnet50 [10]	0.9539	0.9380	0.9413	0.9515	
ShuffleNetV2 [14]	0.9393	0.9040	0.9114	0.9397	
MobileNetV3 [16]	0.9335	0.8881	0.8941	0.9241	
EfficientNet [37]	0.9189	0.8733	0.8790	0.9407	
InceptionV3 [38]	0.9317	0.9519	0.9410	0.9395	
DenseNet [11]	0.9644	0.9380	0.9477	0.9673	
LA-MobileNet	0.9609	0.9515	0.9547	0.9602	

Table 2. Experimental results of different models on the GTOS-mobile8 dataset.

The results in bold mean the best performance under different metrics.

To verify the effectiveness of auxiliary loss and CA, respectively, we conducted experiments, and the results are shown in Table 3. Among them, MobileNetV3+Auxloss and MobileNetV3+CA introduce auxiliary loss and CA into the MobileNetV3 network, respectively, and LA-MobileNet simultaneously introduces auxiliary loss and CA into the MobileNetV3 network. It can be seen that after the introduction of auxiliary loss, Accuracy, Recall, F1-score and Precision increased by 1.80%, 5.05%, 3.63% and 1.98%, respectively, which verified that the auxiliary loss function can provide additional supervisory information during the model training process, help the network learn more complete feature information, and enhance its generalization ability, thereby improving the classification accuracy of the model. After fusing CA, Accuracy, Recall, F1-score and Precision increased by 2.33%, 3.93%, 3.87%, 2.96%, respectively, which verifies that CA can effectively capture the spatial position information in the feature map, improve the model's perception of spatial features, and thereby improve the accuracy of classification. When the above two improvement points are adopted at the same time, the performance of the network is further improved, achieving 96.09% Accuracy, 95.15% Recall, 95.47% F1-score and 96.02% Precision, which proves the effectiveness of the improvements. In addition, the number of parameters of LA-MobileNet is 3.26 M, the floating-point operation amount is 0.31 G, and the time required to predict a single image on our experimental platform is 24.85 ms.

Table 3. Ablation experimental results on GTOS-mobile8 dataset.

Method	Accuracy	Recall	F1-Score	Precision
MobileNetV3	0.9335	0.8881	0.8941	0.9241
MobileNetV3+Auxloss	0.9515	0.9386	0.9304	0.9439
MobileNetV3+CA	0.9568	0.9274	0.9328	0.9537
LA-MobileNet	0.9609	0.9515	0.9547	0.9602

The results in bold mean the best performance under different metrics.

### 4.2. Two-Wheeled Self-Balancing Vehicle Experimental Results

The experiment of forward movement and unilateral obstacle crossing of the wheel-legged balance robot based on the control algorithm is shown in Figure 9. Among them, Figure 9a shows the change in the pitch angle during the advancement of the robot on the horizontal ground. The whole process lasts for 12 s, and the pitch angle fluctuates in the range of  $-2^{\circ} < \varphi < 2^{\circ}$ , which is relatively stable. Figure 9b shows the change in the pitch angle during the unilateral obstacle crossing of the robot. The initial speed of the unilateral obstacle crossing is v = 2 m/s. The whole process lasts 12 s. During the time period of 0–4.92 s, the wheels are traveling on flat ground. The fluctuation range of the pitch angle is  $-2^{\circ} < \varphi < 2^{\circ}$ . At 4.92 s, the single-sided wheel touches the 15° slope, and at 6.21 s, the single-sided wheel breaks away from the 15° slope. The maximum pitch angle change during the entire process is  $-15^{\circ}$  when the wheels are traveling on level ground. After 6.21 s, the pitch angle returns to near the equilibrium point, and parking is achieved at 9 s. The time period of 9–12 s is the change in the upright static equilibrium pitch angle after parking, and the change range is  $-2^{\circ} < \varphi < 2^{\circ}$ .



Figure 9. Experimental results of wheel-legged balancing robot.

#### 5. Conclusions

This paper constructs a stability control system for a two-wheeled self-balancing vehicle in complex terrain. First, a self-balancing vehicle modeling method is designed based on complex terrain and its kinematic model is established to provide a basis for subsequent control algorithms. Secondly, the PID control algorithm is used to adjust parameters such as vehicle attitude, speed and acceleration to achieve smooth operation of the self-balancing vehicle in complex terrain. In addition, in view of the problem that the traditional terrain recognition algorithm model is large and inconvenient to deploy on the mobile terminal, this paper proposes a lightweight terrain recognition network (LA-MobileNet). This network can accurately identify the terrain of the self-balancing vehicle and determine the friction coefficient, thereby ensuring the operational stability of the self-balancing vehicle. LA-MobileNet enhances the representation ability of image features by introducing a coordinate attention mechanism into its backbone network, avoids the loss of position information in two-dimensional global pooling, and constructs an auxiliary loss function to optimize the network. Finally, the effectiveness and stability of the proposed control system in complex terrain are verified through experiments.

**Author Contributions:** Conceptualization, C.Y. and X.L.; methodology, C.Y. and X.L.; case study, C.Y. and X.L.; validation, C.Y.; writing—original draft preparation, C.Y.; writing—review and editing, C.Y.; supervision, X.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the GuangDong Basic and Applied Basic Research Foundation (2022A1515010361).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

**Conflicts of Interest:** Author Chunxiang Yan is employed by the company Guangzhou Automobile Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

# References

- Liu, T.; Zhang, C.; Song, S.; Meng, M.Q.H. Dynamic height balance control for bipedal wheeled robot based on ros-gazebo. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 1875–1880.
- 2. Xin, Y.; Rong, X.; Li, Y.; Li, B.; Chai, H. Movements and balance control of a wheel-leg robot based on uncertainty and disturbance estimation method. *IEEE Access* **2019**, *7*, 133265–133273. [CrossRef]
- 3. Cui, L.; Wang, S.; Zhang, J.; Zhang, D.; Lai, J.; Zheng, Y.; Zhang, Z.; Jiang, Z.P. Learning-based balance control of wheel-legged robots. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7667–7674. [CrossRef]
- Wang, S.; Cui, L.; Zhang, J.; Lai, J.; Zhang, D.; Chen, K.; Zheng, Y.; Zhang, Z.; Jiang, Z.P. Balance control of a novel wheel-legged robot: Design and experiments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 6782–6788.
- Zhang, C.; Liu, T.; Song, S.; Meng, M.Q.H. System design and balance control of a bipedal leg-wheeled robot. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 1869–1874.
- Zürn, J.; Burgard, W.; Valada, A. Self-Supervised Visual Terrain Classification from Unsupervised Acoustic Feature Learning. IEEE Trans. Robot. 2021, 37, 466–481. [CrossRef]
- 7. Guan, T.; Kothandaraman, D.; Chandra, R.; Sathyamoorthy, A.J.; Weerakoon, K.; Manocha, D. GA-Nav: Efficient Terrain Segmentation for Robot Navigation in Unstructured Outdoor Environments. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8138–8145. [CrossRef]
- 8. Otsu, K.; Ono, M.; Fuchs, T.J.; Baldwin, I.; Kubota, T. Autonomous Terrain Classification with Co- and Self-Training Approach. *IEEE Robot. Autom. Lett.* **2016**, *1*, 814–819. [CrossRef]
- 9. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.
- 10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
- Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
- 12. Zhang, Y.; Yu, W.; Zhu, D. Terrain feature-aware deep learning network for digital elevation model superresolution. *Isprs J. Photogramm. Remote Sens.* **2022**, *189*, 143–162. [CrossRef]
- Fei, S.; Chen, Y.; Tao, H.; Chen, H. Hexapod Robot Gait Switching Based on Different Wild Terrains. In Proceedings of the 2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS), Xiangtan, China, 12–14 May 2023; pp. 1325–1330. [CrossRef]
- 14. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
- Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586. [CrossRef]
- Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
- 17. Medeiros, V.S.; Jelavic, E.; Bjelonic, M.; Siegwart, R.; Meggiolaro, M.A.; Hutter, M. Trajectory Optimization for Wheeled-Legged Quadrupedal Robots Driving in Challenging Terrain. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4172–4179. [CrossRef]
- 18. Liu, D.; Wang, J.; Shi, D.; He, H.; Zheng, H. Posture Adjustment for a Wheel-Legged Robotic System Via Leg Force Control with Prescribed Transient Performance. *IEEE Trans. Ind. Electron.* **2023**, *70*, 12545–12554. [CrossRef]
- 19. Zheng, C.; Sane, S.; Lee, K.; Kalyanram, V.; Lee, K. ff-WaLTR: Adaptive Wheel-and-Leg Transformable Robot for Versatile Multiterrain Locomotion. *IEEE Trans. Robot.* 2023, *39*, 941–958. [CrossRef]
- 20. Li, J.; Qin, H.; Wang, J.; Li, J. OpenStreetMap-Based Autonomous Navigation for the Four Wheel-Legged Robot Via 3D-Lidar and CCD Camera. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2708–2717. [CrossRef]
- 21. Li, B.; Li, Y.; Rong, X. The visual terrain classification algorithm based on fast neural networks and its application. In Proceedings of the 32nd Chinese Control Conference, Xi'an, China, 26–28 July 2013; pp. 5780–5784.
- 22. Ebadi, F.; Norouzi, M. Road Terrain detection and Classification algorithm based on the Color Feature extraction. In Proceedings of the 2017 Artificial Intelligence and Robotics (IRANOPEN), Qazvin, Iran, 9 April 2017; pp. 139–146. [CrossRef]

- 23. Liu, F.; Ma, X.; Li, X.; Song, R.; Tian, G.; Li, Y. Terrain recognition for outdoor mobile robots. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 4257–4262. [CrossRef]
- 24. Zhang, W.; Chen, Q.; Zhang, W.; He, X. Long-range terrain perception using convolutional neural networks. *Neurocomputing* 2018, 275, 781–787. [CrossRef]
- Liu, H.; Min, Q.; Sun, C.; Zhao, J.; Yang, S.; Hou, B.; Feng, J.; Jiao, L. Terrain classification with Polarimetric SAR based on Deep Sparse Filtering Network. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 64–67. [CrossRef]
- Wei, Y.; Wei, W.; Zhang, Y. EfferDeepNet: An Efficient Semantic Segmentation Method for Outdoor Terrain. *Machines* 2023, 11, 256. [CrossRef]
- 27. Song, P.; Ma, X.; Li, X.; Li, Y. Deep Residual Texture Network for Terrain Recognition. IEEE Access 2019, 7, 90152–90161. [CrossRef]
- Song, K.; Yang, H.; Yin, Z. Multi-Scale Boosting Feature Encoding Network for Texture Recognition. IEEE Trans. Circuits Syst. Video Technol. 2021, 31, 4269–4282. [CrossRef]
- 29. Kurobe, A.; Nakajima, Y.; Kitani, K.; Saito, H. Audio-Visual Self-Supervised Terrain Type Recognition for Ground Mobile Platforms. *IEEE Access* 2021, *9*, 29970–29979. [CrossRef]
- Tran, D.T.; Hoang, N.M.; Loc, N.H.; Truong, Q.T.; Nha, N.T. A Fuzzy LQR PID Control for a Two-Legged Wheel Robot with Uncertainties and Variant Height. J. Robot. Control 2023, 4, 612–620. [CrossRef]
- 31. Li, Z.; Yang, C.; Fan, L. Advanced Control of Wheeled Inverted Pendulum Systems; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
- 32. Merat, F. Introduction to robotics: Mechanics and control. IEEE J. Robot. Autom. 1987, 3, 166–166. [CrossRef]
- Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717. [CrossRef]
- 34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [CrossRef]
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
- 36. Xue, J.; Zhang, H.; Dana, K. Deep Texture Manifold for Ground Terrain Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 558–567. [CrossRef]
- 37. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June, 2019; pp. 6105–6114.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.