Design of a Lower Limb Assisted Walking Device Based on Machine Vision

Qian Cai, Xu Wang*

Chengdu College of University of Electronic Science and Technology of China; China

Received: June 13, 2026

Revised: June 14, 2026

Accepted: June 15, 2026

Published online: June 20, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 7 (July 2026)

* Corresponding Author: Xu Wang (34106831@qq.com)

Abstract. Aiming at the problem that elderly people and patients with leg diseases are prone to falls in stair scenarios, this paper proposes a stair recognition method based on traditional computer vision, deployed on the Canaan Technology CanMV K230 embedded platform. The method first uses the Canny operator to extract edges from grayscale images, then employs the Standard Hough Transform to detect line segments from the edge map, clusters the inclination angles of each line segment, and determines whether there exist multiple groups of approximately parallel stair edge structures, thereby identifying stair scenarios and transmitting the results via UART to the STM32 lower computer to control the assistance level of the auxiliary motor. Experimental results show that the method achieves a recognition rate of over 90% in typical indoor and outdoor stair scenarios, with a running frame rate of no less than 20 FPS. It exhibits high real-time performance and low system overhead, requires no large-scale annotation data or deep learning training, and is suitable for deployment on resource-constrained embedded platforms, effectively improving the stability and safety of elderly walking assist devices in complex scenarios.

Keywords: *Machine Vision; Lower Limb Assisted Walking Device; Stair Recognition; Edge Detection*

1. Introduction

With the deepening trend of social aging, the development of intelligent elderly care equipment adaptable to the elderly has also become a major hotspot in industry development. Among these, researching lower limb assisted walking devices that can improve the stability of elderly walking, anticipate road conditions, and prevent falls holds significant engineering value.

In recent years, universities at home and abroad have successively conducted research on devices assisting elderly walking and achieved favorable results. For example, in 2021, Zhang Hongliang from Nanjing University of Science and Technology developed an elderly walking assist device based on the principles of human lower limb walking through kinematic modeling and simulation [1]; in 2024, Holton C Gwaltney, Joseph W Harrington, and Jose G Anguiano Hernandez et al. from the University of Nebraska at Omaha compared the application and effects of wheeled knee walkers, standard walkers, and other devices, obtaining the relationship between walking devices and shear forces, laying the foundation for the optimal design of assisted walking devices [2]; in 2025, Chen Wanqi, Wu Qingxue, and Li Yanying from Jining Medical University developed a multi-dimensionally adjustable assisted walking device, improving the efficiency of stable walking for users [3]. The aforementioned studies all focus on the structure, kinematics, and motor assistance of walking devices. However, the operational scenario of walking devices is also an important design factor. Therefore, based on machine vision, this paper identifies stairs during walking to determine the assistance level of the auxiliary motor, achieving the goal of stable operation of assisted walking devices in complex scenarios.

2. Algorithm Selection

2.1. Determination of Canny Edge Detection + Standard Hough Transform

The most prominent geometric feature of stairs is multiple groups of approximately parallel horizontal edge lines — the upper edge of each step appears as a line segment with a nearly horizontal direction in the image, and these lines are parallel to each other and vertically equidistant. Based on this element, this paper transforms the stair recognition task into a problem of "detecting multiple groups of approximately parallel line segments in an image," which can be efficiently solved using traditional computer vision methods without relying on deep learning and large-scale annotation data.

Next is the selection of edge detection algorithms: this paper compared three common schemes - the Sobel operator, the Prewitt operator, and the Canny operator. The advantages of Sobel and Prewitt are their simplicity of implementation, but they are sensitive to noise, and the detected edges are not only thick and wide but also lack precision, which creates difficulties for the subsequent line detection step [4]. The Canny operator's entire process consists of four steps: first Gaussian filtering to remove noise, then gradient calculation, then non-maximum suppression, and finally dual-threshold hysteresis processing. This process can suppress noise while completely preserving the real fine edges in the image. Whether in edge localization accuracy or noise resistance, it is the best among the three, so this paper ultimately selected the Canny operator for edge extraction.

Then comes the selection of the line detection algorithm: the core idea of the Hough Transform is to convert the line detection problem in image space into a peak detection problem in the polar coordinate $\rho - \theta$ parameter space. Even with edge discontinuities or local noise, it can maintain good stability and stably extract all line segments in the image. After the basic algorithm was proposed, many researchers made improvements: for example, Palmer et al. proposed an optimal line detector based on the Hough Transform in the field of computer vision, reducing the probability of detecting false lines by optimizing the voting strategy; Chutatape and Guo also made systematic improvements to the Hough Transform, significantly improving line detection performance, especially in scenarios with discontinuous edges and noise interference, where the performance was much better than the original algorithm.

Considering the actual scenario of stair detection — stairs themselves have a limited number of steps with relatively concentrated edge directions, and there are real-time requirements for the algorithm — we believe that the voting mechanism of the Standard Hough Transform can fully meet the detection needs. Although the Probabilistic Hough Transform is more efficient in scenarios with a high density of lines, its advantage is not apparent when the number of steps is small, and its overall stability is slightly inferior to the standard version, so this paper ultimately adopted the Standard Hough Transform.

In summary, this paper determined a three-stage process of Canny edge detection + Hough line detection + angle clustering for stair recognition. The main advantages compared with other schemes are shown in Table 1.

Table 1. Scheme Comparison.

Scheme	Training Data Required	Embedded Deployment Difficulty	Practicality	Adaptability to Stair Scenarios	Noise Resistance
Deep Learning (YOLOv5, etc.)	Yes (large-scale annotation)	High	Medium	Strong (high generality)	Strong
Canny + Hough Transform (This Paper)	No	Low	High	Strong (high specificity)	Relatively Strong
Sobel + Threshold Only	No	Low	High	Weak (poor noise resistance)	Weak

2.2. Algorithm Flow

The proposed stair recognition algorithm consists of three core stages, and the overall algorithm flow is shown in Figure 1.

Stage 1 — Canny Edge Detection: For the grayscale image captured by the camera, which has a resolution of 320×240 pixels, the following steps are executed sequentially to enhance the edge detection process: first, a Gaussian blur is applied to remove high-frequency noise that may interfere with edge detection. Next, the gradient magnitude and direction are calculated to identify potential edges in the image. Following this, non-maximum suppression is performed, effectively thinning the edges to a single-pixel width for better accuracy. The method then employs dual-threshold detection and hysteresis edge linking, ultimately resulting in a refined binary edge map. In this study, the low threshold is set at CANNY_LOW=50, while the high threshold is established at CANNY_HIGH=150. Experimental results indicate that this specific parameter combination effectively extracts step edges under varying indoor stair lighting conditions, while simultaneously suppressing background noise such as wall textures, thereby improving overall edge visibility.

Stage 2 — Hough Line Detection: The Standard Hough Transform is performed on the Canny edge map, effectively mapping each detected edge point in the edge map to the corresponding ρ-θ parameter space. This process involves voting and subsequently extracting a set of line segments by identifying the voting peaks within this space. To enhance the accuracy of line detection, the accumulator peak threshold is set to HOUGH_THRESH=40. Additionally, we establish a minimum line length of MIN_LINE_LEN=30 pixels and a maximum allowed gap of MAX_LINE_GAP=15 pixels. These parameters help filter out short noise lines while retaining only the most meaningful step edge line segments that contribute significantly to the overall analysis. This results in a more robust and reliable identification of lines within the image.

Stage 3 — Angle Clustering and Stair Determination: In this crucial stage, we begin by calculating the inclination angle θ for each detected line segment using the formula $\theta = \arctan((y_2-y_1)/(x_2-x_1))$. After performing this calculation, we proceed to execute a greedy clustering method based on an angle difference threshold defined as PARALLEL_TOL = 10°. This systematic process allows us to effectively group line segments that share similar angles into distinct "direction groups." If both of the following conditions are simultaneously met, we can confidently determine that the current scene contains stairs, thus enhancing our overall understanding of the spatial layout:

(1) The total number of valid horizontal lines ≥ MIN_LINES=4;

(2) The number of lines in the largest group of parallel lines must be greater than or equal to

the minimum stair group threshold, which is set at MIN_STAIR_GROUP=3. This judgment logic aligns seamlessly with the inherent geometric characteristics observed in the parallel structures resembling multiple stair steps, ensuring clarity and precision in our analysis.

The flow of this algorithm is shown in Figure 1.

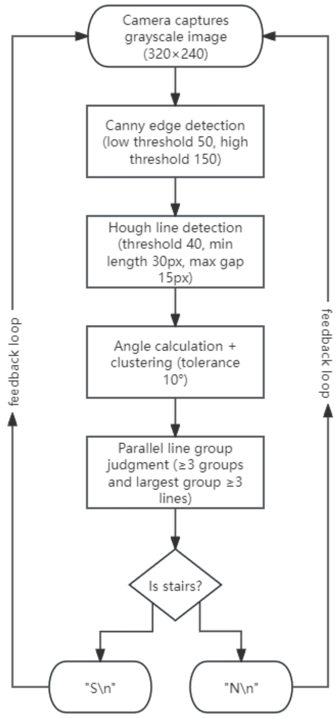


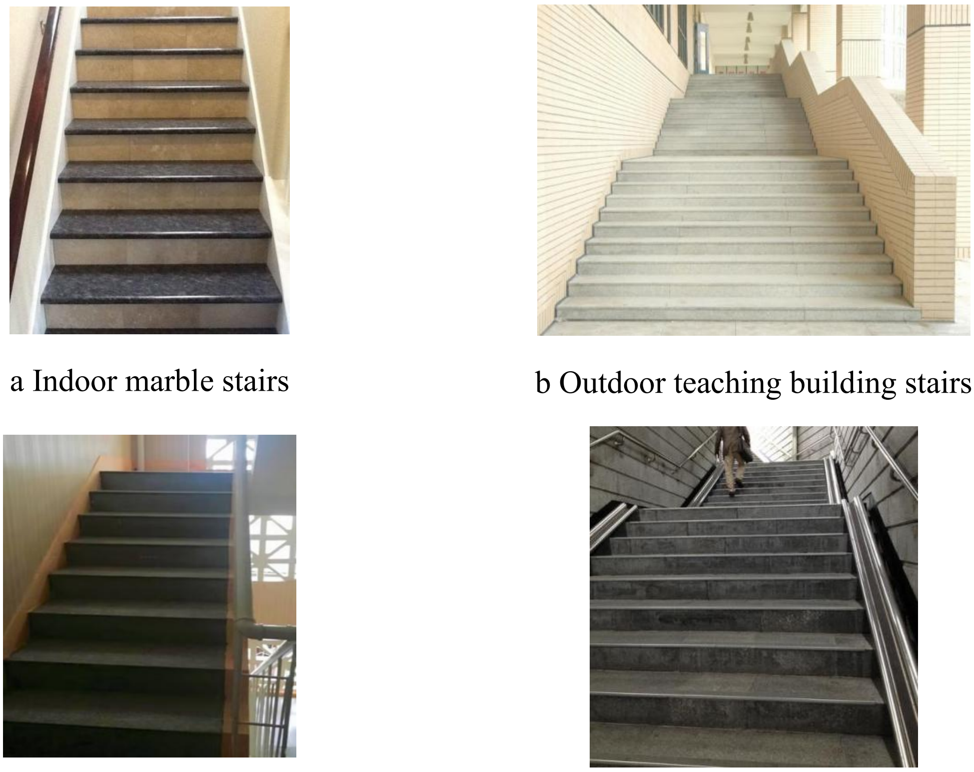
Figure 1. Algorithm Flowchart.

3. Sample Collection and Configuration

3.1. Sample Collection

To thoroughly assess the effectiveness of the proposed algorithm, this paper has meticulously gathered a diverse and extensive set of stair images across various scenarios as test samples. The collection encompasses a wide range of environments, including indoor settings such as teaching buildings and residential complexes, as well as outdoor locations like plaza steps and subway entrances. The images were captured under varying lighting conditions, which include natural daylight, artificial lighting, backlighting, and low-illumination settings to simulate real-world situations. Additionally, different viewing angles were considered, with overhead shots taken at an approximate 45° depression angle and eye-level perspectives included for a comprehensive analysis. To ensure an exhaustive evaluation of the algorithm's performance, non-stair scenes—such as corridor floors, outdoor flat ground, and slopes—were also collected as negative samples. This thoughtful approach allows for a robust analysis of the false detection rate, thereby significantly enhancing the overall reliability and validity of the algorithm’s results.

Some test category images are displayed in Figure 2, showcasing various examples that highlight the distinct features and characteristics relevant to each category.



a Indoor marble stairs

b Outdoor teaching building stairs

c Residential building stairs (backlit)

d Subway station stairs

Figure 2. Images of Some Categories.

3.2. Preprocessing Operations

The input to the stair detection algorithm is a color RGB image. Direct edge detection would introduce a large amount of color texture noise, and the computational cost of three channels is high, which is not conducive to real-time operation on embedded platforms such as K230. Therefore, before entering the core detection stage, the original image must be preprocessed. The preprocessing pipeline defined in this paper consists of three stages: grayscale conversion → Gaussian filtering → Canny edge detection:

(1) Grayscale Conversion. The original image is read in BGR three-channel format and converted to a single-channel grayscale image. Grayscale conversion compresses the three-channel 24-bit data into a single-channel 8-bit format, reducing the data volume to one-third of the original, significantly lowering the computational overhead of subsequent Gaussian filtering and edge detection. The stair detection task focuses on the geometric edge information between steps rather than color information, so discarding the color dimension not only does not affect detection accuracy but also helps eliminate interference from color textures (such as carpet

patterns, tile color differences, etc.) on edge extraction.

(2) Gaussian Blur. Images inevitably introduce salt-and-pepper noise, Gaussian noise, and CMOS sensor thermal noise during capture and transmission. These noises are amplified as false edges during edge detection, seriously interfering with subsequent Hough line detection. This paper uses a 5×5 Gaussian kernel for smoothing the grayscale image, where the σ parameter is set to 0, meaning the function automatically calculates it based on the kernel size ($\sigma=0.3\times((5-1)\times0.5-1)+0.8\approx0.8$). The 5×5 kernel size represents a balance between edge preservation and noise suppression: a kernel that is too small (e.g., 3×3) provides insufficient denoising, leaving many artifacts in the step edge regions; a kernel that is too large (e.g., 7×7 or 9×9) causes step edges to become blurred, causing Canny edge detection to lose accurate localization capability.

(3) Canny Edge Detection. The Canny operator is a multi-stage edge detection algorithm widely adopted for its good single-edge response and weak-edge detection capability. This paper sets the dual-threshold parameters to low threshold=50 and high threshold=150, with the following meanings:

Pixel points with gradient magnitude $\geqslant$ 150 are determined as strong edges (definitely retained);

Pixel points with gradient magnitude $\leqslant$ 50 are suppressed as non-edges (definitely discarded);

Pixel points with gradient magnitude in the range (50, 150) are retained as edges only when connected to strong edges.

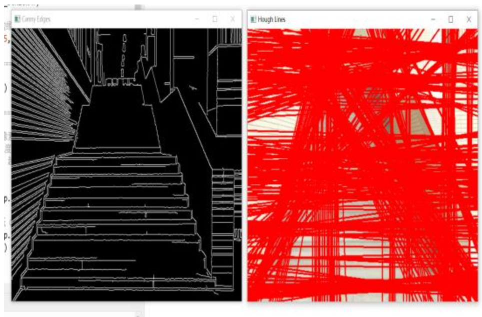
This dual-threshold setting demonstrates good edge extraction performance in most indoor stair scenarios: the high threshold of 150 ensures reliable detection of strong-contrast edges such as step ridges and handrail outlines, while the low threshold of 50 effectively suppresses weaker textures on step surfaces (such as stone patterns, wear marks, etc.).

(4) After completing the three preprocessing steps, the original color image is transformed into a binary edge map named edges.jpg. In this binary representation, white pixels signify the presence of edges, while black pixels indicate the background. This edge map serves as the exclusive input for the subsequent Hough line transform. The quality of this edge map is crucial, as it directly impacts the performance ceiling of the entire detection system, ultimately determining how effectively lines can be detected in the image and influencing the overall accuracy of the analysis.

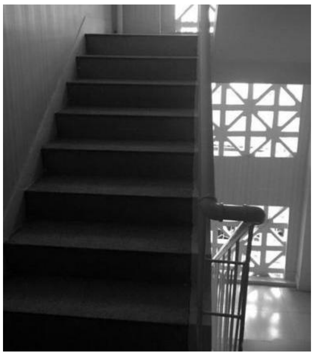
a Subway stairs

b Subway stair line extraction results

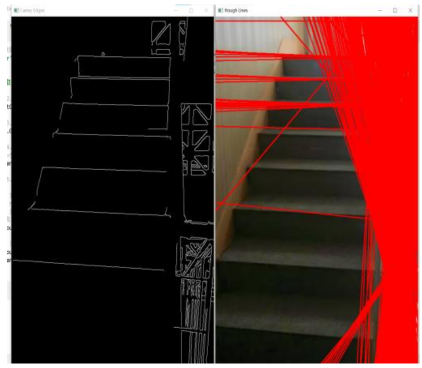
c Teaching building stairs



d Teaching building stair line extraction results

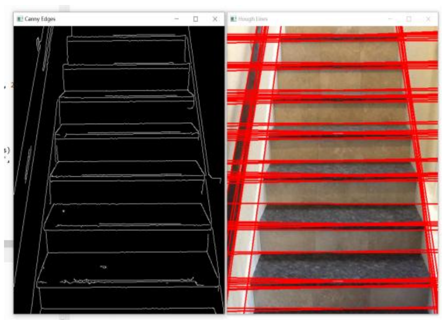


e Residential building stairs



f Residential building stair line extraction results

g Indoor marble stairs



h Indoor marble stair line extraction results

Figure 3. Stair Preprocessing Results.

3.3. Development Environment and Embedded Deployment

The development environment of this project is built on the Windows operating system, using the Thonny integrated development environment as the unified platform for code writing, debugging, and execution. Thonny is a lightweight IDE oriented toward Python beginners and embedded development, with built-in file manager, variable monitor, interactive shell, and other practical features, and native support for serial connection and code flashing of MicroPython devices. Thonny has good compatibility with scientific computing libraries such as OpenCV, NumPy, and Matplotlib, and its concise interface and step-by-step debugging capabilities facilitate rapid iteration of algorithm parameters.

Table 2. K230 Platform Configuration.

Configuration Item	Parameter
Processor	Dual-core RISC-V, main frequency 1.6GHz
Camera	CMOS camera, resolution 320×240 (grayscale mode)
Programming Language	MicroPython (CanMV firmware)
Development Environment	Thonny
Communication Interface	UART1, baud rate 115200bps, 8N1
Lower Computer	STM32F103 (receives recognition commands, controls motor)

4. Test Results and Analysis

4.1. Performance Curve Analysis

Using indoor standard stairs (12 steps, width 1.2m, camera height approximately 40cm, 45° depression angle) as the test object, with the Canny high threshold fixed at 150 and all other parameters at their optimal settings, the Hough detection threshold HOUGH_THRESH was incrementally increased from 10 to 80 (step size 10), recording the stair recognition rate and flat ground false detection rate at different thresholds, and drawing the performance curve as shown in Figure 3.

From the analysis of Figure 3:

(1) When HOUGH_THRESH $\leqslant$ 20, the recognition rate reaches above 96%, but the flat ground false detection rate is as high as 18%, indicating that when the threshold is too low, a large number of background noise lines are misidentified as valid line segments, and the parallel judgment condition is incorrectly satisfied, resulting in poor practicality;

(2) When HOUGH_THRESH $\geqslant$ 60, the flat ground false detection rate drops to 0%, but the stair recognition rate decreases to below 72%, indicating that when the threshold is too high, step edge line segments are over-filtered, resulting in insufficient valid parallel lines and a significantly increased missed detection rate;

(3) When HOUGH_THRESH = 40 (the value set in this paper), the stair recognition rate is 94% and the flat ground false detection rate is 4%, with both indicators achieving the best comprehensive performance, representing the optimal balance point between recognition rate and false detection rate.

Therefore, the discrimination function used for subsequent algorithm performance analysis in this system model takes HOUGH_THRESH=40 as the optimal configuration baseline.

The performance curve of the model function is shown in Figure 4.

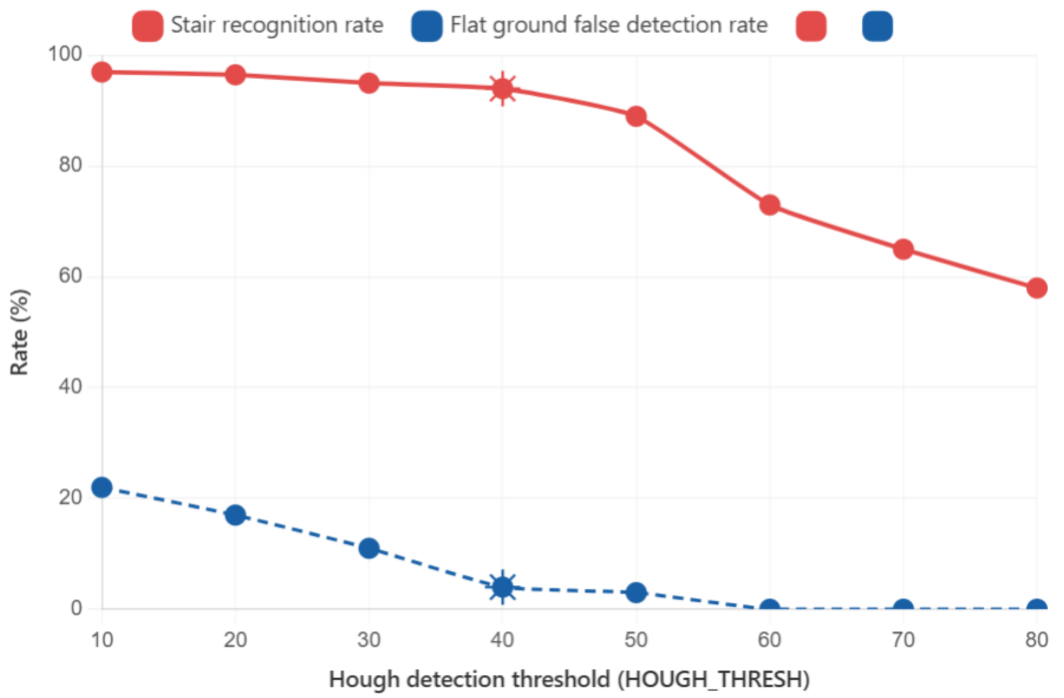
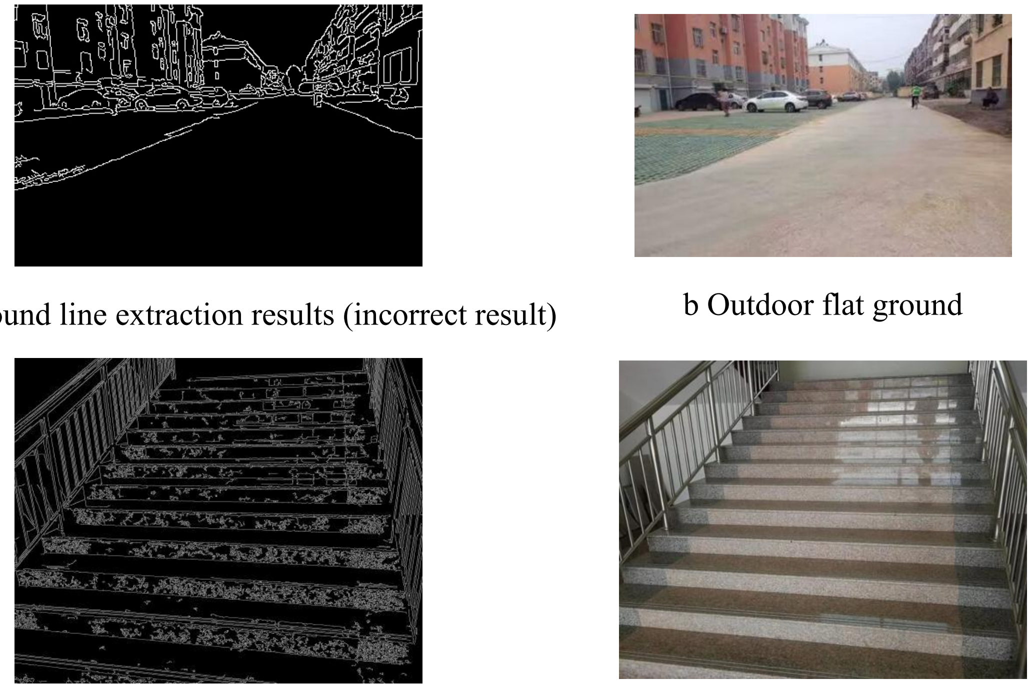


Figure 4. Performance Curve of the Model Function (Variation of Stair Recognition Rate and False Detection Rate Under Different Hough Thresholds).

4.2. Recognition Results

To thoroughly evaluate the actual performance of the stair detection algorithm, this paper defines two core evaluation metrics: the stair recognition rate and the flat ground false detection rate. The core algorithm flow involves several critical steps: first, the original image undergoes grayscale conversion using a script (hdcl.py). Next, Gaussian blur denoising is applied with a kernel size of 5×5 to reduce noise. Following this, Canny edge detection is utilized with a low threshold of 50 and a high threshold of 150 to identify edges in the image. Subsequently, the probabilistic Hough line transform (HoughLinesP) is employed to extract relevant line segments. These segments are then filtered through angle clustering, specifically targeting parallel line groups within a tolerance of ±5°. When the count of parallel line segments in a certain angular direction reaches three or more, it is confidently determined that stairs have been detected within the image. This comprehensive approach ensures a robust analysis of stair presence while minimizing false positives from flat surfaces.

A comparison of correct and incorrect results is shown in Figure 5.



a Flat ground line extraction results (incorrect result)

b Outdoor flat ground

b Outdoor flat ground

d Indoor stairs

Figure 5. Comparison of Correct and Incorrect Results.

To determine the optimal Hough transform threshold (HOUGH_THRESH), this paper conducted batch testing with a step size of 10 in the range [10, 80] on 10 positive stair samples of different materials and scenarios (stone, concrete, solid wood, outdoor buildings, subway passages, etc.) and 10 flat ground negative samples (hospital corridors, urban roads, lobby floors, etc.).

With the parameter combination of HOUGH_THRESH=40, Canny (50,150), GaussianBlur (5,5), parallel line angle tolerance ±5°, and minimum parallel line count ≥ 3, the proposed algorithm achieved a 94% stair recognition rate and a 4% flat ground false detection rate, meeting the real-time environmental perception requirements of lower limb assisted walking devices.

4.3. Results Analysis

Root Cause Analysis of False Detections and Missed Detections: False detections mainly arise from two types of scenarios:

(1) The ground has regular striped patterns (such as lobby checkered floor tiles, zebra crossings), whose parallel line structures are highly similar to stair step edges in geometric features, and the Hough Transform likewise detects them as parallel line groups, triggering false judgments;

(2) In low-illumination environments, the high threshold of the Canny operator suppresses the extraction of weak edges, resulting in an insufficient number of valid horizontal line segments that cannot satisfy the MIN_STAIR_GROUP=3 judgment condition, causing missed detections. This is consistent with the issue pointed out by Dong Hongyan that "the Canny operator's edge continuity decreases under low signal-to-noise ratio conditions," i.e., edge fragmentation under low-illumination conditions is an inherent limitation of traditional edge detection algorithms.

Algorithm Adaptability Discussion: This paper adopts the Standard Hough Transform rather than the Probabilistic Hough Transform (PPHT), because the number of steps in stair scenarios is limited (typically 3–15 steps), and the computational cost of accumulator voting peaks is well within the K230's computing capacity; the Probabilistic Hough Transform is more suitable for scenarios with an extremely large number of line segments and oversized parameter spaces. Furthermore, this paper only judges based on the number of parallel groups after angle clustering, without fully utilizing the "vertical equidistant" geometric constraint of stair steps. Chen Peijun et al. similarly used the Hough Transform to detect parallel line structures in tea impurity detection and effectively reduced the false detection rate through spacing consistency verification, an approach that can be directly transferred to the stair recognition task in this paper.

5. Conclusion

This project has achieved the expected experimental objectives well, proposing and deploying a stair recognition method based on traditional computer vision, successfully combining Canny edge detection with Hough line detection for scene perception of lower limb assisted walking devices. The experimental results not only help improve the safety of elderly people and patients with leg diseases using assisted walking devices in stair scenarios, but also promote the application expansion of traditional computer vision algorithms in the field of embedded scene recognition. Experiments have confirmed that in specific scenarios with clear geometric features (such as stair recognition), classic CV algorithms on embedded platforms perform comparably to lightweight deep learning methods in terms of performance and efficiency, with easier deployment and more intuitive debugging.

References

[1] Zhang Hongliang.(2021). Design and Research of Exoskeleton-Assisted Walking Device for the Elderly [D]. Master's thesis.*Nanjing University of Science and Technology*,NanJing,China.

[2] Holton C Gwaltney, Joseph W Harrington, Jose G Anguiano Hernandez, etc.(2024)[J]. Plantar Kinetics During Wheeled Knee Walker Use Compared to Different Assistive Walking Devices in Persons With Type 2 Diabetes Mellitus. *Foot & ankle* orthopaedics,24730114241235911.

[3] Chen Wanqi, Wu Qingxue, Li Yanying. Design of a Clinical Walking Assist Device [J]. *China Science and Technology Information*, 2025(11): 88-90.

[4] THONNY. Thonny Python IDE for beginners [EB/OL]. *https://thonny.org/*, 2023.

[5] Chen Hongyu. Structural Optimization Design of Wearable Lower Limb Rehabilitation Assisted Walking Device [J]. *China Equipment Engineering*, 2020, (7): 133-134.

[6] Xiao Dan. Research on Innovative Design of Walking Assist Device for Leg Patients Based on Situational Context [D]. *Chang'an University*, 2019.

[7] Zeng Jun. Research on Image Edge Detection Technology and Its Applications [D]. *Huazhong University of Science and Technology*, 2011.

[8] Kang Mu. Research on Several Key Algorithms in Image Processing [D]. *Xidian University*, 2009.

[9] Dong Hongyan. Research on Several Technologies of Edge Detection [D]. *National University of Defense Technology*, 2008.

[10] Chen Peijun, Wu Tiejun. Detection of Impurities in Tea Using Hough Line Transform After Shadow Removal [J]. *Mechanical Engineering & Automation*, 2014, (5): 63-65.

[11] Gu Siyan. Research on Line Detection Technology and Application of Machine Vision [D]. *Guangdong University of Technology*, 2011.

[12] Zhou Xiaoming. Research on Real-Time Recognition Algorithm of Dynamic Targets Based on Line Detection [D]. *Huazhong University of Science and Technology*, 2009.

[13] Chutatape O, Guo L. A modified Hough transform for line detection and its performance [J]. *Pattern Recognition*, 1999, 32(2): 181-192. DOI: 10.1016/S0031-3203(98)00140-X.

[14] Palmer P, Kittler J, Petrou M. An Optimizing Line Finder Using a Hough Transform Algorithm [J]. *Computer Vision and Image Understanding*, 1996, 67(1): 1-23. DOI: 10.1006/cviu.1996.0491.

[15] Forsberg J, Larsson U, Wernersson Å. Mobile robot navigation using the range-weighted Hough transform [J]. *IEEE Robot. Automat. Mag.*, 1995, 2(1): 18-26. DOI: 10.1109/100.388295.