www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Journal of Science and Engineering
Vol: 10, No: 3, 2023 (571-585)
DOI: 10.31202/ecjse.1272769

ECJSE

Research Paper

Design and Study of an AI-Supported Autonomous Stair Climbing Robot

Montaser N. A. RAMADAN [1,a], Shadi M S HILLES[2,b], Mohammad ALKHEDHER [c,3]

[1] Computer Engineering Department, Istanbul OKAN University, Istanbul, Turkiye
[2] Software Engineering Department, Istanbul OKAN University, Istanbul, Turkiye
[3] Mechanical and Industrial Engineering Department, Abu Dhabi University, Abu Dhabi, UAE
montaser.ramadan@gmail.com

Received: 28.03.2023 **Accepted:** 24.08.2023

Abstract: Mobile robots are frequently utilized in the surveillance sector for both industrial and military purposes. The ability to navigate stairs is crucial for carrying out surveillance jobs like urban search and rescue operations. The research paper shows that the design methodology for a six-wheeled rover robot that can adapt to various stairs and maintain its stability based on the robot's specifications, kinematics restrictions, the maximum height, and the lowest step length needed to climb up and down the stairs is proposed. Based on a Raspberry Pi, camera, and LIDAR distance sensor, the suggested robot has the capacity to measure the stair height before starting to climb. A Convolutional Neural Networks (CNN) deep learning model is developed for the purpose of stair recognition. Additionally, stair alignment was estimated using statistical filtering on pictures and LIDAR distance reading. The robot can then decide whether it can climb the stairs or not based on its kinematics limitations and the height of the stairs as measured by our system. Result shows that our stair detection algorithm achieved an accuracy of 99.46% and a mean average precision of 99.64%. The proposed AI- supported Robot-based stair recognition system, according to final results, effectively climbed stairs with a height range between 13 and 23 cm.

Keywords: Raspberry pi, LIDAR, CNN, Robot, AI

1. Introduction

In recent years, mobile robots equipped with cameras and other sensors have grown increasingly ubiquitous, particularly in the surveillance sector. This is due to the fact that it provides options for dynamic monitoring and cost-effective data collecting. The only route to approach a building with numerous stories for unmanned ground vehicles (UGV) is via stairs. But the presence of stairs inside a building presents a significant obstacle to mobile robots' ability to navigate on their own. They are less mobile and able to help in situations like search and rescue [1], disaster mitigation [2], and interior service applications [3] if they are unable to climb stairs. Many different modes of operation, from fully manual [4] to fully autonomous [5] modes, are proposed in the stair climbing literature. In the manual method, the operator's skill plays a role, as does the possibility of making a mistake. Distracting the operator from mission-critical tasks like planning and rescue is counterproductive. The reliance on sensor feedback also makes it susceptible to delays and errors. However, a system that operates without human intervention is both quicker and more reliable than its manual counterpart. Mechanical design, stair detection, orientation, and route planning are just a few examples of how the challenge of autonomous multi-story monitoring might be broken down into more manageable chunks.

The locomotion system is essential in regards to the design of the robot, which is influenced by the environment and technical variables including its mobility and stability [6]. One of a robot's

How to cite this article
M. Ramadan, S. H. Hilles, M. Alkhedher "Design and Study of an AI-Supported Autonomous Stair Climbing Robot," *El-Cezerî Journal of Science and Engineering*, Vol: 10, No: 3, 2023 pp. 571-585.
ORCID: [a]0000-0003-3134-7085; [b]0000-0002-2605-9524; [c]0000-0003-3134-7085

independent tasks is to sense its surroundings and build a controller that corresponds to its pose and location. Robots with wheels are unique in that their stability is maintained by keeping the wheels in contact with the ground [7]. However, because the surfaces have diverse shapes, such as steps and uneven terrain, the robot may fall if its wheels lose touch with the ground. As a result, real-time surface geometry detection will enable the robot to operate fast to preserve stability. The contact angle between the wheel and the ground is a crucial factor in determining the wheels' positions in reference to the surface [8]. To what degree the robot's wheel is climbing a vertical, sloped, or uneven terrain can be determined by this parameter.

The locomotion system plays a crucial role in designing robots, and it is influenced by environmental factors and technical variables like mobility and stability. Robots with wheels maintain stability by keeping their wheels in contact with the ground. However, when encountering surfaces with diverse shapes such as steps or uneven terrain, the robot may lose contact and fall. Real-time surface geometry detection is essential to enable the robot to operate swiftly while preserving stability. The contact angle between the wheel and the ground becomes a crucial parameter for determining the wheel's position relative to the surface. This parameter provides information about the extent to which the robot's wheel is climbing a vertical, sloped, or uneven terrain. By considering such factors, researchers aim to develop robots capable of navigating and climbing stairs autonomously, thus expanding their capabilities for multi-story monitoring tasks.

2. Related works

Many novel locomotive systems have been developed and applied to the creation of stair-climbing robots. They can walk on two legs or go along on wheels. Climbing stairs is typically a mechanical challenge for robots. The authors of [6] explain how they elevate a three-wheeled robot by using a traction control algorithm separately to each wheel. However, it is restricted to stairs that are a specific height. Chenglong Fu and Ken Chen proposed a 32-DOF biped robot that can climb stairs via a feedforward locomotion system [7]. However, it can't be used in a setting where it hasn't been tested. A tracked robot has an advantage over the aforementioned technologies when it comes to the task of stair climbing due to its higher contact surface area, which aids in stabilization. The work of [8] is one of the existing prototypes of tracked stair climbing robots; it uses an online control approach for stairclimbing and force analysis of each stage to achieve maximum capability and stability on the stairs. The ability to climb stairs is primarily dependent on the detection-pipeline and its ability to align with stairs. Using Hough-lines and stereo-vision, the authors of [9] described a monocular method that can be used by the partially sighted to recognize stairs and curbs. In [11], authors used RANSAC-based line fitting for UGV steering and Gabor filters for reliable detection. These conventional methods, though, are built around handcrafted components. Instead of depending on human intuition for feature selection, deep learning approaches enable features to be trained using a dataset. Furthermore, the usage of very deep Convolutional Networks has been made possible and encouraged by improvements in the hardware acceleration for tensor computation (such as GPUs and TPUs). Using region proposals with CNNs (R-CNN)[12] and the sliding windows technique with CNNs (YOLO)[13] have improved the precision and speed of real-time object identification. YOLOv3[14] boasts that it is just as accurate as R-CNN[12] but is a thousand times faster (if not more). Because of its shorter runtime, YOLOv3 was chosen for real-time stair recognition.

3. Robot Design Methodology

3.1. Robot Dimensions and Design Considerations

The design of the mobile robot comprises of a chassis, a Depth-AI camera, a motorized pan/tilt head, and six wheels, each of which can be controlled in their own right by an actuator.

As cleared in Fig. 1. both the front wheel W1 and the middle wheel W2 are connected to a passive rotation joint that is denoted by the bogie and Link L4. This joint has the shape of a triangle that is formed by the parameters L1, L2, and L4. The L1 value indicates how far out from the bogie's center the front wheel is. The middle wheel's center and the bogie's center are separated by a distance, denoted by L2. L4 is the distance between the front wheel to the center wheel. Rocker is responsible for connecting the rear wheel W3, and this wheel maintains a constant distance L5 from wheel 2. It has seen in Fig. 2, the method possesses two passive joints j1 and j2. Because of these joints, all six of the robot's wheels are in constant touch with the ground. One of the joints is a free revolute joint j2, and it may be found in the space between the front and rear Bogie. The second one is referred to as the Rocker pivot j1, and it can be found in the center of L7.

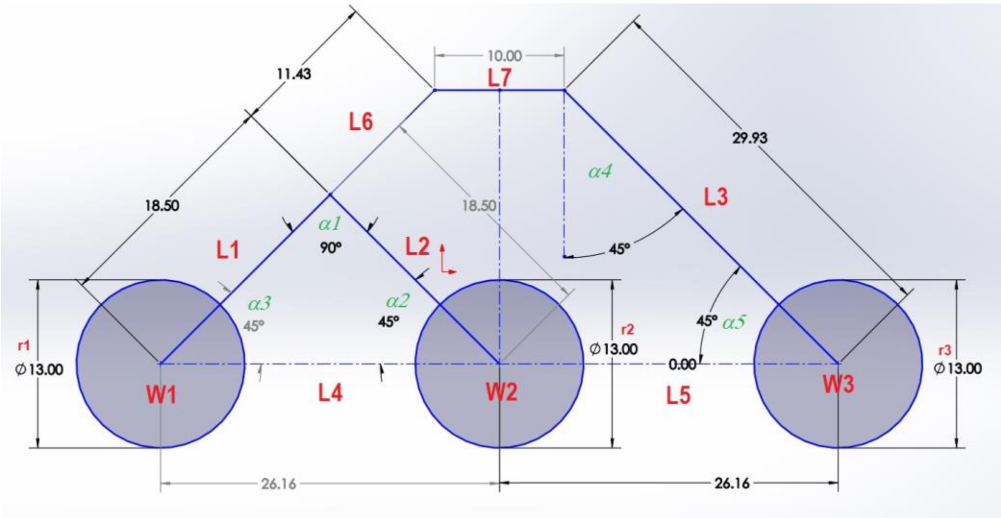

Figure 1. The robot's 2D-schema layout.

Table 1 provides specific information regarding the size and parameters of the robot. Also, we are going to display the angle that is going to be fixed in place between the front and rear Bogie while the robot is in operation. In addition to this, we will discuss the angles that exist between the Rocker and link 7.

Table 1. Robot's rolling chassis settings. Measured in centimeters

Parameters	**Values (cm)**
$l1$	18.5
$l2$	18.5
$l3$	29.93
$l4$	26.16
$l5$	26.16
$l6$	11.43
$l7$	10
$d1$	13
$d2$	13
$d3$	13
$\theta1$	90
$\theta2$	45
$\theta3$	45
4	45
$\theta5$	45

3.2. Prototype Robot Construction Model

The chassis of the robot is made out of a carbon fiber tube that is 25 millimeters in diameter; this not only makes the robot more durable but also helps to keep its weight down. PLA filament is fed through a 3D printer to create each and every joint on the robot. The LiDAR sensor has been mounted in the front of Control Box in order to determine the distance that separates the robot from the stairs. The DepthAI camera is utilized to take photographs at a rate of 30 frames per second, and the Raspberry pi, which is housed within the Control Box, will process each frame individually in order to identify staircases. As demonstrated in Fig. 2, the position of the camera can be adjusted by the use of a motorized pan-and-tilt base.

Figure 2. Model of the robot being developed as a prototype

3.3. Determining the Maximum Climbing Height

The robot's maximum step height was determined by analyzing its kinematic limitations. First, we should calculate l4, we have a right triangle with angle 90 degree between l1 and l2, as shown on Fig. 3.

Figure 3. Max Hight of stairs calculation

First is to calculate:

$$l1 = l2 \ , \theta = 90°$$

$$l4 = \sqrt{l1^2 + l2^2} \tag{1}$$

$$l4 = 26.16 \ cm.$$

$$max. h = r2 + (l4 - r1) * sin \, sin \, \theta \tag{2}$$

$$max. h = 6.5 + (26.16 - 6.5) * 0.89$$

$$max. h = 24 \, cm.$$

It shows that the highest point a robot can climb, that information to determine whether or not our robot will be able to climb stairs. The robot will compare the calculated stair height with the setpoint from equation 2 after we use a deep learning technique to calculate the stair height.

3.4. Stair Detection

3.4.1. Overview

Two primary components of computer vision classification and object detection. Finding what is in an image is called classification, while locating an object in an image is called object detection and localization. Finding the coordinates of the object in a picture makes detection a more difficult challenge to tackle. Deep Learning's capacity to handle enormous volumes of data has demonstrated that it is a very potent technology. Particularly in pattern recognition, hidden layer methods outperform their traditional counterparts. When it comes to Deep Neural Networks (CNN), convolutional networks are among the most popular [15]. CNN is a specific kind of Artificial Neural Network (ANN) utilized for processing data in picture recognition and processing (pixels).

3.4.2. Stair Detection Using CNN Model

Stair detection presented a CNN model and computer vision-based stair to identify new method, in addition, presented subsection address. The deep learning field has access to a variety of enormous datasets that can be used to train models. However, a Robot dataset for stair detections wasn't available till lately. In this work, we used the publicly available StairNet dataset (A Computer Vision Dataset for Stair Recognition) published in 2022 [16] to train and evaluate our proposed detector. The StairNet Dataset is a sizable hierarchical database of photos captured by wearable cameras and is specifically designed to identify incline staircases. Every image in this dataset was taken from ExoNet and categorized [17]. A total of 515,452 photos make up the collection, which is divided into four classes: 442,360 level ground (LG) images, 15,888 level ground incline stairs (LGIS), 9025 incline stairs level ground (ISLG), and 48179 incline stairs (IS). To train, test, and validate our CNN-based detector, we employed a total of 5000 different Stairs and Non-stair images. The resolution of the used images is 1280x720 pixels. StairNet dataset examples are shown in Fig. 4.

There are 500 unannotated photos in the testing set. The photos have features in common with the training set. Some, but not all, of the testing set's locations are also present in the training set. As shown in Table 2, the images from the training and testing datasets are nearly evenly distributed across the two classes, Stair and Non-Stair.

Figure 4. The dataset for StairNet is composed of. Stair class example image on the left. Right: An example of a non-stair class

Table 2. Distribution of the dataset for stair detection

Class	Data Size
Stair	2500
Non-Stair	2500

We employ deep learning to estimate the function f that connects the visual features or image from our input I[in] to the desired discrete output y[n]. Two potential outcomes are chosen for each y[n] label (i.e., stair or no-stair). Instead of using a preexisting model, this research makes use of Neural Architecture Search to generate a convolutional neural network model that optimizes the accuracy performance criterion. So to sense, we make use of:

$$f^*, w^* = \ argmin_{f,w}[\sum_{I[n],y[n]\in T} L(f(I[n];w), y[n])\ ] \tag{3}$$

where T is the collection of training picture data, which represents a randomly chosen 80 percent of all data. Finding the best weight is just one of our goals; we also want to identify the best model architecture (e.g., number of layers, their types, their connections, and their hyperparameters).We predict the stair detection model using our data using a recurrent neural network (RNN). Based on the validation set D accuracy, which is considered as 10% of the data and the remaining 10% for testing and performance evaluation, the incentive signal used to train the RNN is determined. A controller RNN network and a child convolutional neural network are the two networks in use to determine the ideal weight and model design. The CNN produced is then used to find the probability of stair detection, or

$$p^+[n], p^-[n] = \ f^*(I[n]; w^*) \tag{4}$$

The architecture of the stair detection model is described in Table 3. This model requires an RGB color image with a 256x256 pixel input size. According to Table 3, each input image travels through a number of convolution layers, with the first layer beginning with 16 convolution kernels that are 3x3 with a stride of 2. Additionally, convolutional blocks are arranged as three-layer bottleneck blocks with several filters and a stride of 1 as shown in Fig. 5.A completely connected layer is placed at the bottom of the network to perform the stair detection, and each unit in the final layer is connected to two output probabilities using the Softmax function. As a result, the suggested detector generates the probabilities of positive class detection, also known as "Stair," and negative class detection, also known as "Non-Stair," which are denoted by the numbers $p^+[n]$, and $p^-[n]$ respectively.

Table 3. Information on the proposed CNN-based stair detector

Layer Name	Output Size	Kernel
conv2d_21 (Conv2D)	254x254x16	3x3x3x16
max_pooling2d_18	127x127x16	
conv2d_22	125x125x32	3x3x16x32
max_pooling2d_19	62x62x32	
conv2d_23	60x60x16	3x3x16x32
max_pooling2d_20	30x30x16	
flatten_6 (Flatten)	14400	
dense_18 (Dense)	256	14400x256
dense_19 (Dense)	1	256x1

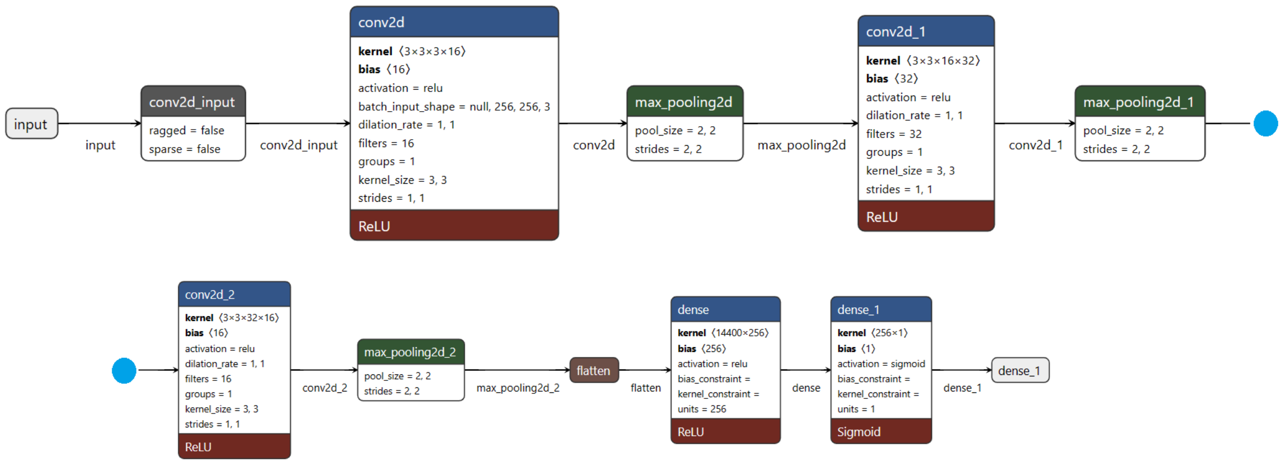

Figure 5. Change capture to Proposed CNN model

We employ the test-set cross validation method, and the training, testing, and validation sets that we used have 4493, 561, and 561 images, respectively. We choose to use the ratio of 80:20 for the ratio of training to testing dataset, as indicated in Table 4.

Table 4. Data distribution

Data Type	Training	Validation	Testing
Percentage	80%	20%	
		10%	10%

4. Results and Discussion

We assess the performance of the suggested stair detector in terms of accuracy, precision, recall, and F1 measure. Table 5. illustrates the general confusion matrix that sums up the detector's performance.

Table 5. Confusion Matrix Labels

		Predicted Label	
		Stair	Non-Stair
True Label	Stair	TP	FN
	Non-Stair	FP	TN

As indicated in Table 5, we define TP, TN, FP, and FN as follows:

- True Positive (TP): The proposed stair detector correctly identified a stair when one was present.
- True Negative (TN): The proposed stair detector indicated that there would be no stairs, but none were there.
- False Positive (FP): The suggested stair detector incorrectly identified the scene as having stairs when none were there.
- False Negative (FN): The proposed stair detector indicated that there was no stair in the scene when there was actually one.

We can get the performance measurements for any of the detectors using the aforementioned variables. Accuracy is the percentage of stairs that are accurately detected calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

Precision is the ratio of correctly positive stair detections to all correctly positive detector predictions. It can be calculated using:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Recall is the percentage of accurate positive stair detections to all accurate detector predictions. Recall is a measure of the detector's stair prediction sensitivity, where:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

Finally, the F1-Score is the most indicative statistic because it represents a harmonic mean between recall and precision.

$$F1 - Score = 2\,x\,\frac{Recall\;xPrecision}{Recall\;+Precision} \tag{8}$$

In order to quantitatively validate our proposed detector, we trained it on 5,054 photos split into "Stair" and "Non-Stair" classes. On a total of 561 images distributed evenly between the "Stair" and "Non-Stair" classes, for this evaluation, we employ an IoU threshold of 0.5 to determine how well the detector performs.

We calculate the performance metrics and tabulate them in Table 7 using Equations 3, 4, 5, and 6. The proposed detector attained a 99.46% detection accuracy and a 99.64% mean average precision.

Table 6 shows the confusion matrix of the detection model based on the labels presented in Table 5.

		Predicted Label	
		Stair	Non-Stair
True Label	Stair	99.2%	0.8%
	Non-Stair	0.3%	99.7%

Table 7. Stair detection performance metrics

Performance Measure	Score
Accuracy	99.46%
Precision	99.64%
Recall	99.29%
F1-Score	99.46%

4.1. Calculating the Height of the Stair by Computer Vision

After the robot detects the stair based on proposed CNN model, the next step is to calculate the height of the stairs automatically based on computer vision. The stair detection frame will be processed Realtime by five major steps as shown in Fig. 6.

- First: To streamline the procedure and reduce the amount of processing power required, we transform the color image to a grayscale one.
- Second: We used a low-pass Gaussian Blue filter to get rid of the high-frequency noise and set the level of sharpness. And that important to reduce the noise as much as possible.
- Third: Then, to identify numerous edges in images, the well-known Canny Edge Detection technique is used.
- Fourth: After the image was smoothed, the imperfections were minimized using dilation by increasing the size of the foreground convolutional area.
- Fifth: After contours were extracted from the edge picture, we removed all the extraneous detection points.

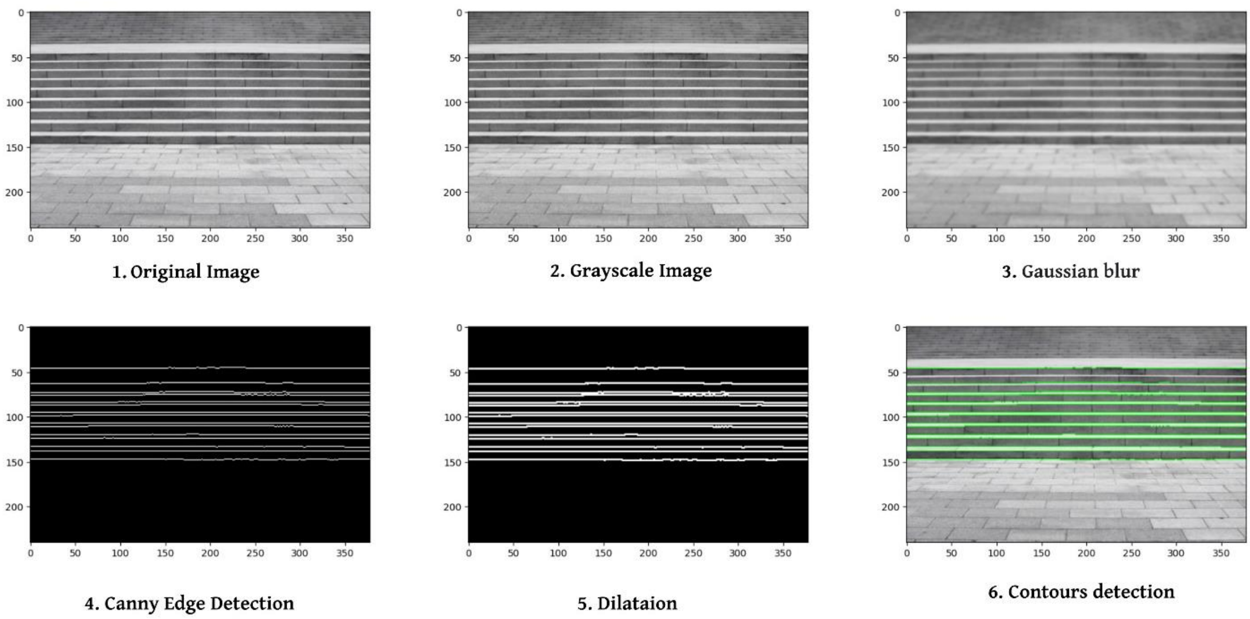

Figure 6. Stair detection process

To calculate the height of the stairs, the following equation should be used:

$$h = \frac{H\ x\ D}{F} \tag{9}$$

where:
h: The stair's height in centimeters
H : Pixel value representing the stair's height
D : The distance between camera and stair (measured by LiDAR sensor).
F: The camera's focal length.

The camera's focal length must be established first, so let's position an object in front of it that is a certain height, and fix the camera at a known distance from the object, as shown in Fig. 7.

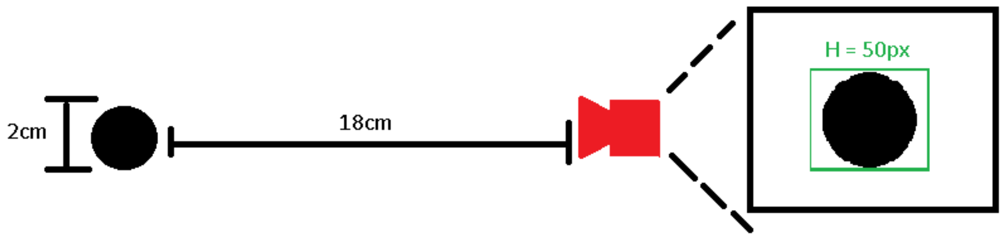

Figure 7. a fixed camera in front of a known-distance object

Returning to equation 7, the formula below should be used to determine the camera's focus length:

$$F = \frac{H\ x\ D}{h} \tag{10}$$

$$F = \frac{50px\ x\ 18cm}{2cm} = 450 \tag{11}$$

The camera's focal length is known. Since the height of the stair in pixels is known from the image and the distance between the camera and the stair is measured using a LiDAR sensor, equation 7 may be used to get the actual height of the stair.

4.2. Climbing stairs of varying heights

Several tests involving the ascent of stairs with heights between 13 and 23 centimeters were conducted. Fig. 8. shows the stairs that were used to train the suggested robot. The robot was instructed to climb a set of stairs that ranged in Hight from 13 to 16 centimeters in the experiment's first phase. Following that, the robot was able to complete the second experiment, which involved climbing a stairway that was 23 centimeters tall, without incident.

4.2.1. The first set of experiments

The initial testing began with stairs that ranged in height from 13 to 16 cm. In the beginning, the camera continuously captured frames and processed them using the suggested CNN model for stair detection. Following the robot's detection of the stairs, it began calculating the stair heights using a computer vision algorithm that followed the various steps depicted in Fig. 9. First, a color image was converted to a grayscale one in order to streamline the process and use less processing power. then the level of sharpness was set and the high-frequency noise was removed using a low-pass Gaussian Blue filter.

Figure 8. Various stair heights used in our tests

And it's crucial to minimize noise as much as possible. After that, the well-known Canny Edge Detection method was applied to identify numerous edges in the images. Following image smoothing, the blemishes were reduced using dilation by enlarging the foreground convolutional area. All the extraneous detection points were eventually eliminated after contours were extracted from the edge image.

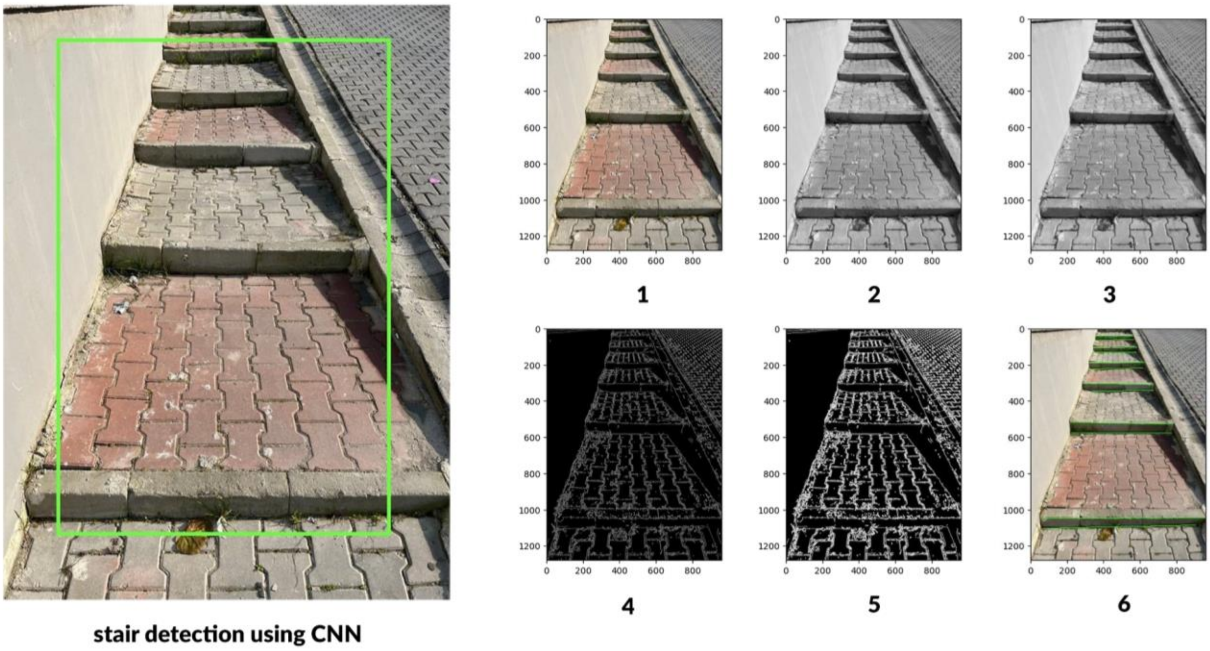

Figure 9. CNN model and computer vision algorithm output before climbing stairs

Following the stairs' detection using the suggested CNN model, equation 11 was used to calculate the stairs' height, which came out to be 14 cm. After determining that the stairs' height fell within its operating range, the robot began climbing the stairs as shown in Fig. 10. Multiple tests were conducted at the same set of stairs, and each time the robot successfully claimed more than 10 stairs with Hight ranged between 13 to 16 cm.

Figure 10. The robot was able to ascend stairs that ranged in height from 13 to 16 cm

4.2.2. The Second Set of Experiments

In this part of the study, the robot's capacity to climb stairs of varied heights was put to the test. At first, the robot was able to effectively ascend steps ranging in height from 13 to 16 centimeters (cm). Nevertheless, we intended to test the capability of the robot to ascend and descend steps of varying heights.

In order to present the robot with a greater task, stairs with a height more than 20 centimeters were implemented. The robot was put to the test, and it successfully displayed its capability by climbing a set of steps that were 23 centimeters in height as shown in Fig. 11. This accomplishment demonstrates the robot's better climbing capabilities and hints that it may be able to manage stairs with large height disparities in the future.

Figure 11. Our robot scaled a 23 cm-high stair.

The achievement of these results represents a significant achievement in the development of the stair-climbing capabilities of the robot. The capability of the robot to climb steps of diverse heights, including those that are higher than 20 centimeters, broadens the range of applications for which it may be used and improves the possibility that it will be able to navigate different kinds of surroundings.

5. Conclusions

In conclusion, the methodology for the design of a six-wheeled rover robot is proposed in the research work that is capable of adapting to various stairs and maintaining its stability, making it ideal for surveillance jobs such as urban search and rescue operations. The robot is equipped with a Raspberry Pi, camera, and LIDAR distance sensor, which allow it to measure the stair height before starting to climb. In order to estimate stair alignment, a prototype robot with a depth-AI camera and six actuator-controlled wheels was created. The climbing height was determined by examining the kinematic constraints. For the purpose of stair detection, the system employs a deep learning model based on convolutional neural networks (CNN). The CNN model-based StairNet dataset of a robot for stair detection, was identified and examined. The performance of the stair detector in terms of accuracy, precision, and recall has been computed. Additionally, real-time stair detection frames using image enhancement based on filters like low pass gaussian blue and canny filters has been proposed and tested. The results indicate that the stair detection algorithm achieved an accuracy of 99.46% and a mean average precision of 99.64%. The proposed AI- supported Robot-based stair recognition system was able to effectively climb stairs with a height range between 13 and 23 cm, demonstrating its potential for use in industrial and military surveillance applications.

Future Studies

In the future studies on mobile robots and autonomous navigation in multi-story buildings are likely to focus on several key areas.

Advanced Perception and Sensing: Researchers will explore improved methods for perception and sensing in mobile robots to enhance their ability to detect and navigate stairs based on developing more sophisticated camera systems, and integration additional sensors.

The adaptive locomotion systems is also one of future studies and it may investigate novel locomotion systems that enable robots to navigate stairs and uneven surfaces with increased stability and agility, included machine learning and AI techniques to improve the autonomy and decision-making capabilities of mobile robots in stair climbing scenarios and could involve training models to recognize and adapt to different stair configurations, optimize navigation paths, or handle unexpected obstacles or changes in the environment, in addition, as autonomous mobile robots become more prevalent in real-world scenarios, future studies may focus on optimizing human-robot interaction during multi-story monitoring tasks and could involve designing intuitive interfaces for operators to supervise and control robots, developing collaborative systems where humans and robots work together in a coordinated manner, and as well to ensuring the robustness and reliability of stair-climbing robots will continue to be a significant research focus. Future studies may investigate fault-tolerant mechanisms, redundant systems, or resilience strategies that allow robots to adapt to failures or unexpected conditions during stair climbing operations.

Authors’ Contributions

MR led the methodology, hardware and software development, validation, original draft writing, review & editing. MA played a role in conceptualization, methodology design, investigation, resource management, original draft writing, review & editing, supervision. SH made substantial contributions to the investigation, resource management, data curation, and review & editing processes.

Competing Interests

The authors declare that they have no competing interests.

References

[1]. D. Jennings and M. Figliozzi, "Study of Sidewalk Autonomous Delivery Robots and Their Potential Impacts on Freight Efficiency and Travel," *Transportation Research Record*, vol. 2673, no. 6, pp. 317-326, 2019.

[2]. T. Hoffmann, G. Prause, "On the Regulatory Framework for Last-Mile Delivery Robots," *Machines*, vol. 6, no.3, pp.33. 2018.

[3]. U. Patil et al., "Deep Learning Based Stair Detection and Statistical Image Filtering for Autonomous Stair Climbing," *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, 2019, pp. 159-166, doi: 10.1109/IRC.2019.00031.

[4]. PT. T Nguyen., S. W. Yan, J. F. Liao, C. H. Kuo "Autonomous Mobile Robot Navigation in Sparse LiDAR Feature Environments," *Applied Sciences*, vol. 11 no. 13, pp. 5963, 2021. https://doi.org/10.3390/app11135963

[5]. A. Li, J. Cao, S. Li, Z. Huang, J. Wang, G. Liu, "Map Construction and Path Planning Method for a Mobile Robot Based on Multi-Sensor Information Fusion," *Applied Sciences*, vol. 12 no. 6, pp. 2931, 2022.

[6]. S. S. Samsani and M. S. Muhammad, "Socially Compliant Robot Navigation in Crowded Environment by Human Behavior Resemblance Using Deep Reinforcement Learning," *in IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5223-5230, July 2021, doi: 10.1109/LRA.2021.3071954.

[7]. S. V. Sreenivasan, and B. H. Wilcox, "Stability and traction control of an actively actuated micro-rover," *Journal of Robotic Systems*, vol. 11, pp. 487-502, 1994. https://doi.org/10.1002/rob.4620110604

[8]. K. Nagatani, A. Ikeda, K. Sato and K. Yoshida, "Accurate estimation of drawbar pull of wheeled mobile robots traversing sandy terrain using built-in force sensor array wheel," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 2373-2378, doi: 10.1109/IROS.2009.5354566.

[9]. M. Frigola, A. Casals and J. Amat, "Human-Robot Interaction Based on a Sensitive Bumper Skin," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 283-287, doi: 10.1109/IROS.2006.282139.

[10]. H. Xu, X.Liu, H. Fu, et al, "Visual Contact Angle Estimation and Traction Control for Mobile Robot in Rough-Terrain," *Journal of Intelligent & Robotic Systems*, vol. 74, pp. 985-997, 2014.

[11]. N. Pico, Hr. Jung, J. Medrano, "Climbing control of autonomous mobile robot with estimation of wheel slip and wheel-ground contact angle," *Journal of Mechanical Science and Technology* vol.36, pp.959-968, 2022.

[12]. H. Cevallos, G. Intriago and D. Plaza, "Performance of the Estimators Weighted Least Square, Extended Kalman Filter, and the Particle Filter in the Dynamic Estimation of State Variables of Electrical Power Systems," *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, Concepcion, Chile, 2018, pp. 1-6, doi: 10.1109/ICA-ACCA.2018.8609729.

[13]. Dong Hun Shin and Kyung Hoon Park, "Velocity kinematic modeling for wheeled mobile robots," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Seoul, Korea (South), 2001, vol. 4, pp. 3516-3522, doi: 10.1109/ROBOT.2001.933162.

[14]. T. Thueer, A. Krebs and R. Siegwart, "Comprehensive Locomotion Performance Evaluation of All-Terrain Robots," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 4260-4265, doi: 10.1109/IROS.2006.281954.
[15]. S. Seok et al., “Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot,” *in IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1117-1129, June 2015, doi: 10.1109/TMECH.2014.2339013.
[16]. M. Ramadan, M. Al-Khedher, S. Al-Kheder, “Vehicle Tracking Using An Efficient And Low Cost Embedded System With GPS and GSM”, *Proceedings of the 4th International Conference on Advanced Computer Theory and Engineering (ICACTE 2011)*, Dec. 28-30, 2011, Dubai, UAE.
[17]. Montaser N. A. Ramadan, Shadi M S Hilles, Mohammad Alkhedher, and Mohammed Ghazal, “Real-time Automated License Plate Recognition and Tracking of Runaway Vehicles,” *International Informatics and Software Engineering Conference*, Ankara, Turkey, IEEE Xplore, 2022, pp. 1-5
[18]. M. Ramadan, M. Al-Khedher, S. Al-Kheder, “Intelligent Anti-Theft and Tracking system for Automobiles”, *International Journal of Machine Learning and Computing*, vol. 2, no. 1, pp. 83-88, 2012.
[19]. M. Ramadan, M. Al-Khedher, “Wheelchair Control Using 3 DOF Head Gesture Recognition”, *International Journal of Mechanical and Production Engineering*, vol. 3(4), 2015, pp: 42-47.
[20]. A. Bisher, M. Al-Khedher, “Intelligent mapping and locomotion system for blind guidance robot”, *Proceeding of the 5th International Symposium on Mechatronics and its Applications (ISM08)*, May 27-29, 2008, Amman, Jordan.
[21]. M. Al-khedher, “Integrated Robotic System for Indoor Spatial Perception and Navigation for the Blind”. *International Review on Modelling and Simulations*, vol. 5, no. 2(B), 2012.
[22]. M. Al Khawaldah, M. Al-Khedher, I. Al-Adwan, A. Al Rawashdeh, “An Autonomous Exploration Strategy for Cooperative Mobile Robots”, *Journal of Software Engineering and Applications*, vol. 7, no. 3, pp. 142-149, 2014.
[23]. M. Ghazal, T. Basmaji, M. Yaghi, M. Alkhedher, M. Mahmoud, A. Elbaz, “Cloud-based Monitoring of Thermal Anomalies in Industrial Environments using AI and the Internet of Robotic Things,” *Sensors*, vol. 20, no. 21, pp. 6348, 2020.
[24]. M Ghazal, M Yaghi, A Gad, G Bary, M Alhalabi, M Alkhedher, “AI-Powered Service Robotics for Independent Shopping Experiences by Elderly and Disabled People,” *Applied Sciences*, vol. 11, no. 19, pp.9007, 2021.