Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 16, No. 2, 2025

A Computer Vision-Based Software for Calculating Automotive Wiring Harness Length

Jun Heng [1, 2, 3], Jiale Qian [1, 2, 3], Yanxin Liu [1, 2, 3], Haoyu Fu [1, 2, 3], Yi Zeng [1, 2, 3, *], Lingling Shen [1, 2, 3], Qian Gao [1, 2, 3], Xiaojun Qian [1, 2, 3]

[1] School of Computer and Electronic Information, Nanjing Normal University, Nanjing, China
[2] School of Artificial Intelligence, Nanjing Normal University, Nanjing, China
[3] Artificial Intelligence Research Institute, Nanjing Normal University, Nanjing, China

[*] **Corresponding author**: Zeng Yi (Email: zengyi@njnu.edu.cn)

Abstract: With the rapid development of electrification and intelligence in the automotive industry, the structural complexity of wiring harnesses continues to increase. Traditional methods for measuring wiring harness length face challenges such as low accuracy and slow efficiency when dealing with curled and branched harnesses. To address these challenges, this paper presents the design and implementation of a computer vision-based software solution for accurately calculating automotive wiring harness lengths. The software utilizes a computer vision model to calibrate the start and end points of harness branches, and employs image preprocessing techniques (including binarization and denoising) to obtain clear images of the harness. A skeletonization algorithm is then applied to refine and extract the structural outline of the harness. Based on this, a Depth-First Search (DFS) algorithm is used to accurately calculate the length of each branch. Experimental results show that the software designed in this paper can accurately measure the length of each branch in a short period, with an average accuracy of 94.36%. This method provides effective technical support for intelligent inspection and automated production of automotive wiring harnesses, with significant application value and broad prospects.

Keywords: Computer Vision; Depth-First Search; Wiring Harness Length Calculation.

1. Introduction

With the rapid development of the global automotive industry, the level of electrification and intelligence in vehicles continues to improve. As a core component of automotive electrical systems, automotive wiring harnesses are becoming increasingly complex. The wiring harness is responsible for providing power and signal transmission to various electronic devices in the vehicle, making its design and optimization crucial to ensuring the overall performance and safety of the vehicle.

Currently, the final inspection of automotive wiring harnesses primarily relies on manual operations. Workers need to manually measure the harness by wrapping it around a testing platform, which is time-consuming and labor-intensive, severely affecting production efficiency. Therefore, how to achieve automated and intelligent inspection has become a key issue that needs to be addressed.

In the factory inspection of automotive wiring harnesses, multiple processes are typically required, including component testing, defect detection, continuity testing, and length measurement. Yuan Haibing et al. from Hubei Automotive Industry Institute [1] applied an improved YOLOv7 model for defect detection of harness terminals. Huang Xinkang et al. from Zhejiang Sci-Tech University [2] proposed an image-based method for automotive wiring harness structure inspection, integrating digital image processing technology and deep learning to detect harness components. Wang Yanmei from Changchun University [3] of Technology designed an STM32-based automotive wiring harness testing device for automated continuity testing. Li Wei-Chen et al. from National Taiwan University of Science and Technology [4] used a pattern-matching algorithm to achieve precise detection of wiring harness components.

Abroad, Hao Wang et al. from Chalmers University of Technology [5] used a deep learning model to detect automotive wiring harness connectors, achieving high success rates. Ioana Stefan et al. from the University Politehnica of Bucharest [6] proposed an automated assembly method for wiring harnesses, optimizing the harness assembly process.

In the field of length measurement, Li Xiaobao et al. from Nanjing University of Aeronautics and Astronautics [7] proposed a machine vision-based method for measuring the size of electronic wiring harness components, achieving measurements with an error of ±0.1 mm for shorter harness components. Kamble Supriya S. et al. from College of Engineering, Pune, India [8] designed an optical-based computer vision wiring harness detection system for measuring the length of shorter, straight harnesses.

Regarding component testing, defect detection, and continuity testing, current research generally applies to various forms of wiring harnesses. However, in the area of length measurement, existing research has focused on simpler, shorter harnesses, and there is still a lack of effective solutions for measuring branched, tree-like harnesses.

To address this issue, this paper proposes an automotive wiring harness length automatic measurement software based on computer vision and depth-first search (DFS) technology. The software imports images of the harness and uses advanced image processing algorithms, computer vision techniques, and DFS methods to achieve fast and accurate calculation of the length of each branch. Unlike traditional manual measurement and CAD-based measurement methods, this software does not rely on complex instrumentation but utilizes standard camera equipment and image recognition technology to automate the measurement process. Through this innovative approach, users can obtain accurate wiring harness length data in a short period, significantly improving inspection efficiency and reducing human errors.

This research not only has significant practical application value in the automotive manufacturing process but also provides new technical support for the design and optimization of automotive wiring harnesses. The implementation of this software enhances production efficiency and improves the accuracy of wiring harness inspection, with broad practical application potential and profound theoretical significance.

2. System Architecture

The system architecture of this software is based on a modular design, divided into four main modules: user interface, component detection, image Preprocessing, and length calculation. The system provides a user-friendly interface, allowing users to easily upload images and obtain the calculated results for the lengths of the wiring harness branches. The main flowchart of the system architecture is shown in Figure 1.

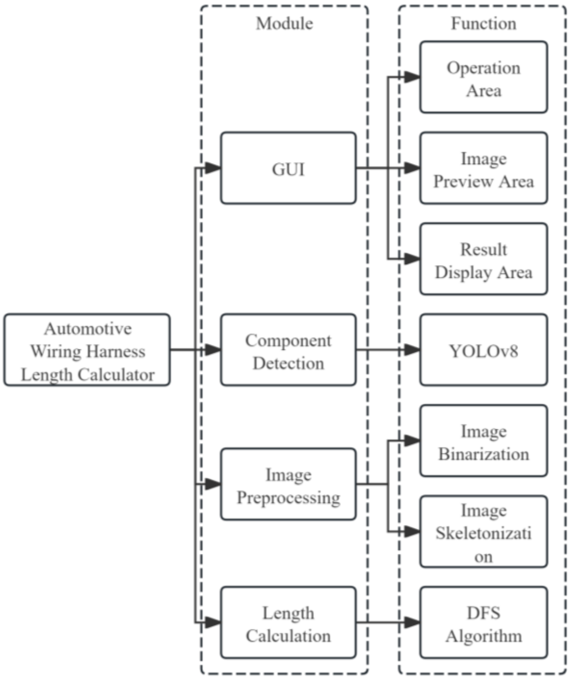

Figure 1. System Architecture Diagram

2.1. Overall Architecture Overview

The overall system architecture can be divided into the following five components:

User Interface: The user uploads the wiring harness image via the graphical interface, initiates the measurement process, views the calculation results, and saves the output data.

Component Detection: The system uses a pre-trained YOLOv8 model to detect the motor and terminals of the wiring harness, calibrating the starting and ending points for length calculation.

Image Preprocessing: The system uses the OpenCV library to preprocess the image, including binarization and skeletonization operations, to facilitate subsequent length calculation.

Length Calculation: The system uses the Depth-First Search (DFS) algorithm to traverse the skeletonized image and compute the length.

2.2. Technology Stack and Algorithms

Programming Language: Python
Image Processing Library: OpenCV
Computer Vision Model: YOLOv8
Database: SQLite

2.2.1. Automotive Wiring Harness Length Calculation Scheme Design

The algorithm studied in this paper is based on the A2LL/A2SL model of wiring harnesses, which has the following characteristics: long lengths, with the longest branch measuring 152.7 cm; multiple branches, with 21 branches connected to a single harness, each with a terminal at the end; and a curled shape. Due to the material properties, the harness typically appears curled in its natural state. The shape of the wiring harness is shown in Figure 2.

Figure 2. Schematic of Wiring Harness

Given the above characteristics of the wiring harness, and drawing upon the application of skeletonization algorithms in plant growth analysis (e.g., fruit trees) [9], this paper first uses a computer vision model to calibrate the starting and ending points for length calculation. A skeletonization algorithm is then used to fit the structural skeleton of the harness, and the Depth-First Search (DFS) algorithm is employed to calculate the length from the starting point (the connection between the motor and the harness) to each terminal of the harness. The design flow is shown in Figure 3.

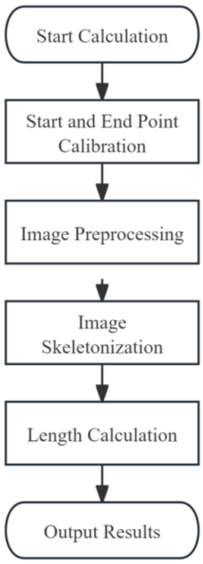

Figure 3. Flowchart illustrating the steps involved in harness length measurement

Once the algorithm is initiated, it first uses the pre-trained computer vision model to detect the motor and terminals of the wiring harness for start and end point calibration. The image is then preprocessed, including binarization, noise reduction, and dilation/erosion operations, to minimize the impact of external factors such as the camera and light source on the harness, and to obtain a clean binary image of the harness. The skeletonization algorithm is then applied to convert the binary harness image into a skeleton with a width of one pixel. Finally, the length of each branch is computed based on the obtained skeleton.

2.2.2. Training of the Computer Vision Model and Result Processing

This paper uses YOLOv8 for component detection. The starting point is set at the connection between the motor and the harness, while the endpoint is set at the connection between each terminal and the harness. This calibration is crucial for accurate length measurement. The model was trained using a dataset consisting of 230 images, with the results summarized in Table 1.

Table 1. Model Training Performance

Model	Precision	Recall
YOLO v8	0.976	0.979

This model can detect the motor and terminals in the harness image and define their locations using bounding boxes. The system then intersects the bounding boxes with the harness to determine the final start and end point positions for length calculation.

2.2.3. Image Preprocessing

The calibrated wiring harness image must undergo preprocessing to reduce the impact of uneven lighting, shadows, and gaps in the shooting platform on subsequent analysis. The goal of preprocessing is to extract a binary image containing a clean background (white base) and foreground (wiring harness body). The specific steps include grayscale conversion, binarization, denoising, and morphological processing, which respectively eliminate lighting fluctuations, highlight the harness body, and remove noise and breaks.

This paper uses the OpenCV open-source library for image preprocessing and subsequent operations. Its powerful image processing functions support the implementation of the method. In this module, functions such as cv2.cvtColor(), cv2.bitwise_not (), cv2.dilate(), and cv2.erode() are primarily used.

First, the original RGB three-channel image is converted to a single-channel grayscale image, reducing the computational load, simplifying the algorithm, and speeding up processing while retaining the original harness length and morphological features.

Then, a fixed threshold method is used to binarize the grayscale image. Since the background of the picture is almost white and the harness body is almost black, a gray threshold (binary_threshold) is set. Pixels above this threshold (close to white) are considered background, and pixels below it (close to black) are considered foreground, resulting in a binary image for subsequent feature extraction.

Next, the outer contours of the binary image are extracted, keeping only those with an area larger than a set threshold (area_threshold) to remove small gaps and other noise from the shooting platform.

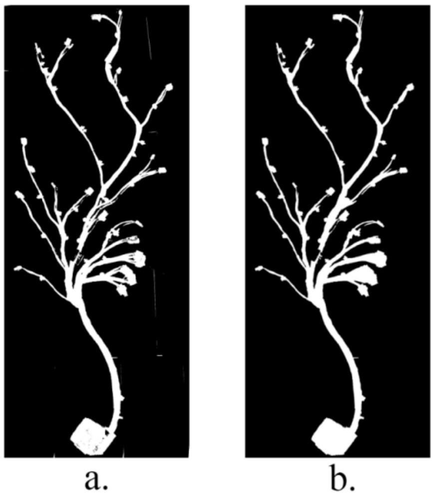

Figure 4. a. Binarized Image Before Denoising, b. Binarized Image After Denoising

Finally, morphological processing is applied to the denoised image to restore the true shape of the wiring harness. Due to the presence of light-colored labels and terminals on the harness, these components may be misidentified as background during binarization, which would interrupt the harness structure. If not addressed, this could prematurely halt the depth search of the entire image. Therefore, the processed binary image undergoes erosion and dilation operations. Erosion is used to remove excess branches, and dilation is used to connect small interruptions. The erosion operation effectively removes noisy pseudo-edges, while dilation reconnects breakpoints and restores the complete skeleton of the harness. Through multiple experiments, the result of applying erosion followed by dilation (Figure 5c) yields better recovery of the harness's overall structure. The accuracy of the erosion and dilation algorithms is determined by the structural elements, typically selected as a (2k+1)*(2k+1) (where k=0,1,2,...), rectangular structural element for processing. After processing, the broken sections of the original binary image are connected.

Figure 5. Illustration of Erosion and Dilation Operations, a. Unprocessed Broken Image, b. Effect of Erosion Followed by Dilation, c. Effect of Dilation Followed by Erosion

2.2.4. Harness Skeletonization

After binarization and denoising, the harness image may still contain redundant width or complex shapes that directly impact the accuracy of the subsequent length calculation. The goal of skeletonization is to compress the harness body into a one-pixel-wide skeleton, retaining its shape and connection features while reducing unnecessary redundant information.

This paper adopts the Zhang-Suen algorithm for the skeletonization operation. The Zhang-Suen algorithm is a classic image thinning method that iteratively removes redundant pixels while preserving the topological structure and morphological features of the image. This algorithm is selected due to its efficiency and stability in extracting the skeleton of harness images. The overall algorithm flow is as follows:

Initialize the flag variable changeFlag=True to determine whether further iterations are needed.

Set the sub-iterator subIteration=1 to distinguish different deletion conditions.

In each iteration, check each pixel in the image to see if it meets the deletion condition (based on 8-neighborhood connectivity and pixel number changes).

If the current pixel meets the deletion condition, mark it for deletion; otherwise, keep it.

After one round of scanning, if no pixels are deleted, stop the iteration and output the final skeletonized image, as shown in Figure 6.

In the wiring harness image, components like terminals and motors are usually large in size. Direct skeletonization could turn these regions into tangled branches, disturbing the subsequent length judgment of branches. Therefore, an adaptive region filtering algorithm is used to remove excess short branches, avoiding their interference with the endpoint determination of length calculation.

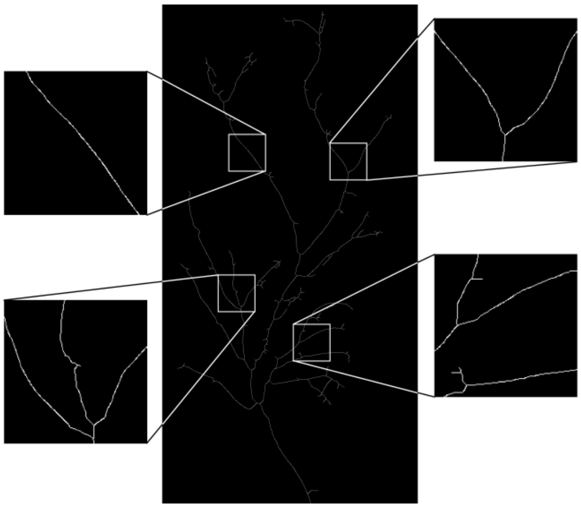

Figure 6. Skeletonization Effect of Wiring Harness

2.2.5. Length Calculation Algorithm

After obtaining the wiring harness skeleton image, the next goal is to calculate the pixel length of each branch to accurately restore the actual length distribution of the harness. Due to the complex multi-branch tree structure of the harness, traditional algorithms struggle to efficiently handle the length calculation of multi-branch and curled harnesses. The depth-first search (DFS)-based length calculation method proposed in this paper traverses each path of the harness skeleton, accurately accumulates the pixel length, and maps it to the actual physical length. The algorithm flow is shown in Figure 7.

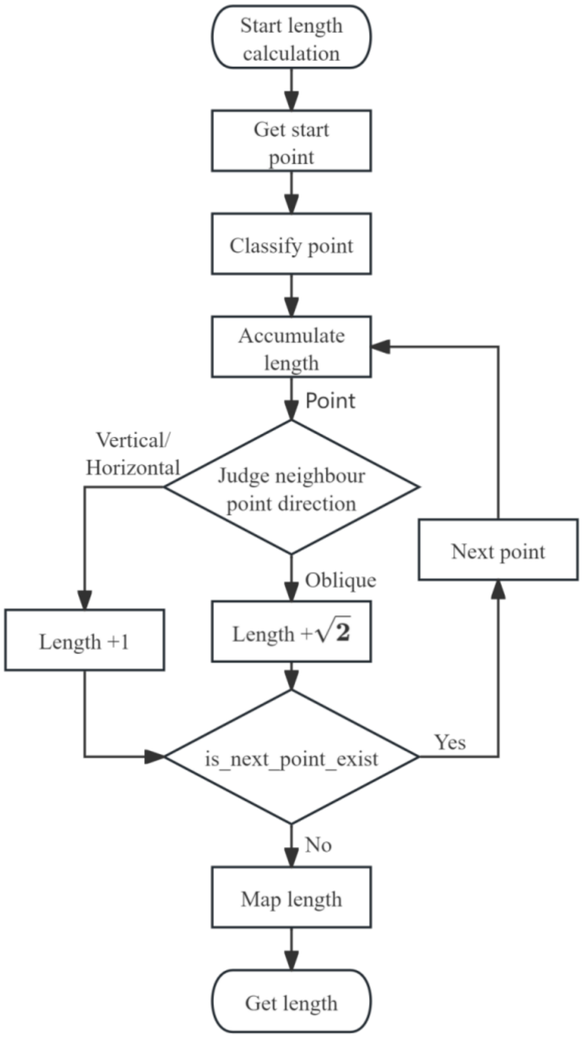

Figure 7. Harness Length Calculation Flowchart

To ensure the accuracy of length calculation, the algorithm first classifies the pixels in the skeleton image to determine the affiliation of each pixel. The classification criteria are as follows:

Endpoint: If the foreground/background pixels switch in a clockwise direction once in the 8-neighborhood.

Branch Point: If the foreground/background pixels switch four or more times in the 8-neighborhood.

Ordinary Point: If the foreground/background pixels switch three times in the 8-neighborhood.

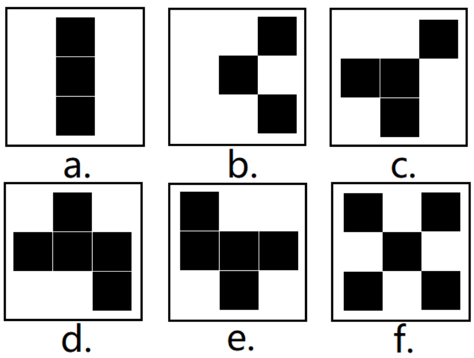

Figure 8. Six Possible Pixel Branch Scenarios

Based on the key point classification results of the skeleton image, the algorithm uses depth-first search (DFS) to traverse all branches of the wiring harness, accumulating pixel lengths. The DFS traversal rules are as follows:

Start Point Selection: From the classified key points, select an unvisited endpoint as the start point and begin the traversal.

Path Extension: Starting from the current pixel, prioritize traversing its 8-neighborhood pixels in a clockwise direction, finding unvisited foreground pixels and marking them as visited.

Branch Point Processing: If a point has multiple unvisited foreground pixels (i.e., branch points), push the current path onto the stack and process each branch until all paths are traversed.

Path Length Accumulation: Each time a pixel is visited, its length is accumulated according to the following rules:

If the pixel is in the horizontal or vertical direction, the length will add 1.

If the pixel is in the diagonal direction, the length will add $\sqrt{2}$. After traversal is complete, the algorithm outputs the pixel length of all branches.

Since the wiring harness skeleton image is a discrete representation in pixels, its length needs to be mapped to the actual physical length through a certain ratio. This paper uses the following formula for length mapping:

$$L_{real} = \frac{L_{pixel}}{R} \quad (1)$$

Where:

L_{real} is the actual length (in cm).

L_{pixel} is the accumulated pixel length in the skeleton image.

R is the pixel density (in pixels/cm), obtained through experimental calibration.

3. Experimental Results and Performance Analysis

The interface for successfully conducting length measurement with the software presented in this article is shown in Figure 9.

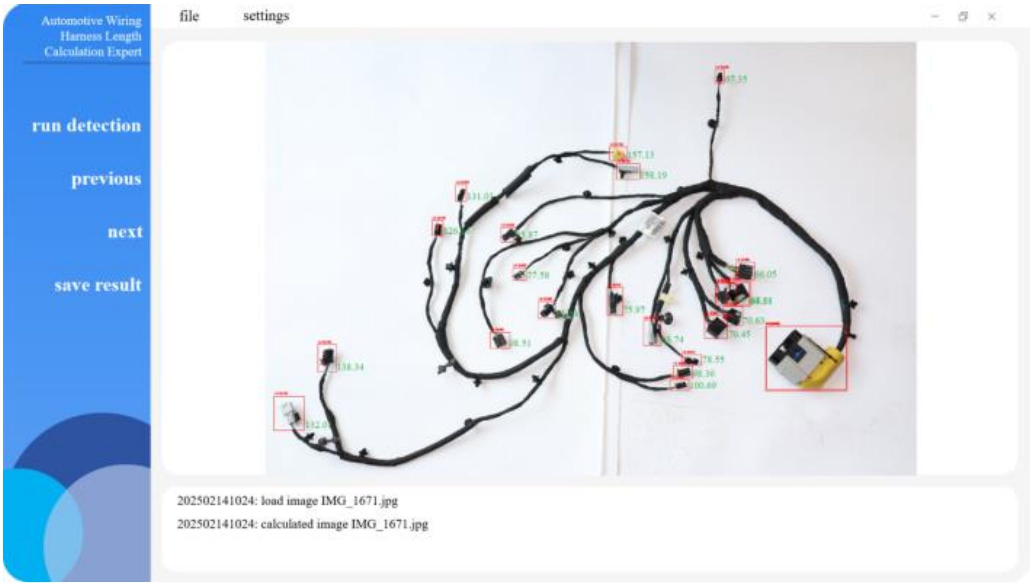

Figure 9. Length Measurement Success Example

Length measurement experiments were performed on 12 images using the software developed in this study. The error results for each branch length are shown in Table 2.

Table 2. Branch Lengths and Error Calculation Results

Branch Number	True Length (cm)	Calculated Length (cm)	Absolute Error (cm)	Relative Error (%)
111-1111	67.1	65.8	4.2	6.23%
111-1112	68.4	65.3	3.1	4.51%
111-1113	67.3	64.0	3.3	4.86%
112-1121	71.1	70.1	1.3	1.88%
112-1122	70.9	69.7	1.8	2.55%
113-1131	83.2	77.3	5.9	7.11%
113-1132	75.3	67.4	7.9	10.52%
1-12	66.4	63.0	3.7	5.51%
2-21	90.9	84.2	6.7	7.36%
22-221	104.2	98.7	5.5	5.26%
22-222	83.7	79.4	4.3	5.11%
22-223	85.8	81.4	3.6	4.27%
3-31	76.4	71.2	5.2	6.83%
4-41	107.1	98.0	9.1	8.48%
4-42	106.3	99.5	6.8	6.42%
5-51	133.2	139.3	6.1	4.56%
5-52	128.3	132.8	4.5	3.54%
6-61	135.6	125.1	10.5	7.77%
7-71	142.4	130.0	12.4	8.73%
8-81	150	157.8	7.8	5.19%
8-82	152.7	155.2	2.8	1.81%

The overall harness error is shown in Table 3:

Table 3. Harness Length Calculation Error

Root Mean Square Error (RMSE)	**6.14cm**
Mean Absolute Percentage Error (MAPE)	5.17%
Longest Branch Percentage Error (LBPE)	1.81%
Shortest Branch Percentage Error (SBPE)	5.51%

The algorithm used in the software achieves a Root Mean Square Error (RMSE) of 6.14 cm when processing branch lengths ranging from 66.4 cm to 152.7 cm. This indicates that the algorithm controls the errors well for most branches. The average relative error is 5.17%, with 90% of the branches having errors below 10%. Further analysis shows that branches with shorter lengths (<70 cm) tend to have larger errors. This could be due to the fact that short branches are more sensitive to variations in pixel density, suggesting that the robustness of the algorithm needs to be improved for shorter branches.

Time and Space Complexity Analysis

In terms of time complexity, the most time-consuming tasks of the algorithm are skeleton extraction and DFS traversal. The time complexity for skeleton extraction is $O(NM)$, where N and M are the width and height of the image resolution. The complexity of DFS traversal is $O(V+E)$, where V and E represent the number of pixels and edges in the skeleton. Experiments show that the algorithm takes an average of 4 seconds to process an image with a resolution of 4K, proving its efficiency in high-resolution scenarios.

In terms of space complexity, the memory consumption of the system is mainly due to the storage of the original image and the intermediate results generated during processing, as well as the stack used during the DFS traversal. Overall, the system has a space complexity of $O(NM)$, where N and M are the width and height of the image resolution.

4. Conclusion

The software designed in this study utilizes computer vision models and digital image processing techniques to measure the lengths of wiring harness branches. The YOLOv8 model is used to calibrate the start and end points of each branch. By applying certain functions from the OpenCV library, the harness images undergo binarization, noise reduction, and thinning, resulting in skeleton images that retain the shape and length features of the harness. Using a depth-first search (DFS) algorithm, the software measures the lengths of the branches in complex, curled harnesses. The system operates with high accuracy and short measurement times, making it valuable for practical production applications. However, the algorithm experiences higher errors in measuring shorter harness branches, and future work will focus on improving this aspect.

Acknowledgment

This work was supported in part by the Huai'an Science and Technology Program for Frontier Technology Research and Development (HAG202416)

References

[1] H. Yuan, F. Zhao, Y. Yang, et al., "Research on wiring harness defect detection based on improved YOLOv7," Foreign Electronic Measurement Technology, vol. 43, no. 2, pp. 165–173, 2024. DOI: 10.19652/j.cnki.femt.2305480.

[2] X. Huang and Y. Yuan, "Image-based automotive wiring harness structural inspection," Computer Era, no. 6, pp. 129–133, 2023. DOI: 10.16644/j.cnki.cn33-1094/tp.2023.06.026.

[3] Y. Wang, Design and implementation of an automotive wiring harness detection device based on STM32, M.S. thesis, Changchun Univ. of Technology, 2023. DOI: 10.27805/d.cnki.gccgy.2023.000678.

[4] W. Lee and K. Cao, "Application of machine vision to inspect a wiring harness," in Proc. 2019 IEEE Int. Conf. Industrial Cyber Physical Systems (ICPS), pp. 457–460, 2019. DOI: 10.1109/ICPHYS.2019.8780292.

[5] H. Wang and B. Johansson, "Deep learning-based connector detection for robotized assembly of automotive wire harnesses," in Proc. 2023 IEEE 19th Int. Conf. Automation Science and Engineering (CASE), pp. 1–8, 2023. DOI: 10.1109/CASE56687.2023.10260619.

[6] I. Stefan, A. Popescu, C. Popa, T. Dobrescu, and C. Cotet, "Research regarding assembly flow optimization of wiring harness in automotive industry," in MATEC Web of Conferences, 2021. DOI: 10.1051/matecconf/202134302003.

[7] X. Li, Y. Lu, L. Zhou, et al., "Electronic wiring harness product dimension detection method based on machine vision," Journal of Testing Technology, vol. 38, no. 3, pp. 289–297, 2024.

[8] S. Supriya and A. Ashwini, "Automatic optical inspection system for wiring harness using computer vision," in Proc. 2021 IEEE Int. Conf. Electronics, Computing, and Communication Technologies (CONECCT), pp. 1–5, 2021. DOI: 10.1109/CONECCT52877.2021.9622654.

[9] W. Zhang, X. Li, H. Wan, et al., "Corn plant stem and leaf segmentation method based on skeleton extraction and binary tree analysis," Henan Agricultural Science, vol. 49, no. 9, pp. 166–172, 2020. DOI: 10.15933/j.cnki.1004-3268.2020.09.021.