Advances in Computer, Signals and Systems (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2025.090103
ISSN 2371-8838 Vol. 9 Num. 1

Design and Implementation of an Intelligent Gesture Recognition and In-Vehicle System Based on OpenCV

Li Jian[1,a,*], Wang Xin[2,b]

[1]Department of Information Technology, Beihua University, Jilin, China
[2]Department of Computing, Beihua University, Jilin, China
[a]434237159@qq.com, [b]1686789636@qq.com
**Corresponding Author*

Keywords: OpenCV; gesture recognition; on-board systems; Interactive

Abstract: With the rapid development of information technology, intelligent gesture recognition technology has been widely applied in the field of human-computer interaction. This paper aims to design and implement an intelligent gesture recognition in-vehicle system based on OpenCV to enhance the safety and convenience during driving. The system captures the driver's gestures through a camera, processes the images using OpenCV, and combines machine learning algorithms to accurately recognize gestures, converting the recognition results into control commands for the on-board system. Experimental results indicate that the system can effectively recognize various gestures and accurately execute the corresponding control operations, demonstrating high practical value.

1. Introduction

With the relentless progression of artificial intelligence technology, recognition technologies have evolved to unprecedented levels of sophistication. Human-computer interaction has transcended the confines of traditional two-dimensional interfaces, such as mice, keyboards, or capacitive screens. Gesture recognition-based human-computer interaction has not only emerged but has also become a pivotal paradigm shift. The significance of realizing gesture interaction within the constraints of low-specification hardware environments is particularly noteworthy, as it paves the way for more accessible and ubiquitous interactive experiences, even in resource-limited settings.

1.1. Background

Human-computer interaction initially evolved with the primary objective of facilitating more convenient control of computers and information input, predominantly through the use of keyboards and mice. However, during that era, users were incapable of interacting with computers using natural gestures, which significantly limited the intuitiveness and fluidity of the interaction process. This limitation necessitated the development of highly automated and sophisticated devices specifically designed to bridge this gap, thereby marking the inception of research in the field of human-computer interaction. The pursuit of more natural and intuitive interaction methods has since driven continuous innovation and advancement in this domain, aiming to create more seamless and user-friendly experiences.

Due to the flexibility and wide range of expressions of the human hand, tracking and recognizing hand movements has become a focal point. The approach taken involves studying the foundational theories before practical operations. In the foundational theory section, current domestic and international research achievements in the field of computer vision are analyzed and discussed to highlight the significance of the research. Subsequently, the basic knowledge of gesture recognition is studied to provide a theoretical basis for the subsequent system design. In the practical application section, an experimental platform is set up to preprocess the acquired gesture images and detect and recognize moving gestures, achieving research applications similar to mouse operations.

1.2. Domestic and International Status

Significant strides have been made in the realm of gesture recognition technology within the domestic sphere, particularly in systems that are predicated on the confluence of computer vision and machine learning algorithms. Numerous research institutions and enterprises are not only actively engaged in research and development but are also at the forefront of introducing innovative solutions. These advancements are geared towards enhancing user experience and operational efficiency, as evidenced by the development of a gesture recognition method based on a three-dimensional hand skeleton model, which has shown high recognition accuracy and real-time response capabilities in various environments. Furthermore, the integration of multimodal technologies, such as eye tracking, is poised to elevate the intelligence level of gesture recognition systems, thereby enriching the user experience. This continuous evolution and application of gesture recognition technology are indicative of its growing significance and potential in a wide array of application scenarios, promising to revolutionize human-computer interaction.

For example, gesture recognition systems based on OpenCV have been widely applied domestically. These systems analyze gestures captured by cameras to achieve natural interaction with computers. Additionally, there are gesture recognition technologies based on data gloves and other sensor systems. Although these are cumbersome to wear and relatively expensive, they still hold value in certain specific fields. Internationally, the development of gesture recognition technology started earlier, and the technology is relatively mature. Many well-known international enterprises and research institutions have made important breakthroughs in this field.

For instance, research teams from Inha University and Korea Polytechnic University in South Korea have used entropy analysis to segment gesture areas from video streams with complex backgrounds and perform gesture recognition, achieving an average recognition rate of over 95%. Moreover, international efforts are also exploring gesture recognition technologies based on advanced algorithms such as deep learning and neural networks to enhance the accuracy and stability of recognition. Although there have been numerous studies on gesture recognition both domestically and internationally, the application of this technology to in-vehicle systems is relatively limited. Based on previous research, this paper designs and implements an intelligent gesture recognition in-vehicle system based on OpenCV, taking into account the characteristics of in-vehicle systems.

2. System Design

2.1. Improved Content

The system of this product employs YOLO image recognition technology and the Qt Creator integrated embedded development environment. It reduces the operational difficulty of devices such as car music players and windshield wipers, providing users with a more simplified and convenient operating environment. The design of the product aims to inspire and create more possibilities for gesture recognition technology and embedded functions in vehicles.

Regarding research on gesture recognition technology, a mature system has been established domestically, capable of detecting the bending index of fingers and spatial positioning index, making three-dimensional interaction more realistic and natural. However, the cost of this technology is inevitably high, and even if it is perfected, it cannot be widely used on a large scale. Therefore, achieving gesture interaction in low-specification hardware environments remains a future trend.

2.2. Research Content and Key Issues to be Addressed

This product develops a gesture recognition music player based on the YOLOv5 object detection algorithm, and subsequently innovates by adding an embedded hardware windshield wiper function. Using YOLOv5 algorithm technology and C++ language, it realizes six basic functions of gesture control for the music player, including play/pause, song switching, lyric display, and volume adjustment. [1]After the basic functions of the lower computer are realized, training is conducted to achieve gesture recognition on the upper computer. The upper computer realizes gesture recognition, while the lower computer realizes the music player function. After multiple refinements, the entire set of gesture recognition-based music players has been designed and implemented. The integration of the control functions of these two modules provides users with more reasonable interaction logic and a more convenient user experience.

2.3. Flowchart of Intelligent Gesture Recognition and In-Vehicle System Based On OpenCV

Figure 1 shown that Flowchart of Intelligent Gesture Recognition and In-Vehicle System Based On OpenCV. The system presented here utilizes OpenCV for intelligent gesture recognition to control a vehicle's infotainment system. This setup allows for a touchless and intuitive user interface, enhancing the driving experience

Camera: A camera is used to capture gesture images. It should be installed inside the vehicle to clearly capture the driver's gestures. Arduino: Serves as the control unit, receiving instructions from the computer and controlling the vehicle's functions. Motors and Actuators: Connected to the Arduino, these components control the vehicle's movements and other functions.

Python: The primary programming language for implementing gesture recognition and control logic. OpenCV: An open-source computer vision library used for image processing and gesture recognition. MediaPipe: A library for hand keypoint detection, enhancing the accuracy of gesture recognition. PySerial: A library for serial communication with Arduino.

Image Capture: The camera captures a real-time video stream, and each frame is sent to the computer for processing. Image Preprocessing: OpenCV is used to preprocess the images, including grayscaling and Gaussian blurring, to reduce noise and enhance gesture features. Gesture Detection: MediaPipe detects hand keypoints to identify the position and posture of the hand. OpenCV's contour detection and feature matching techniques are used to further confirm the shape and action of the gesture. Gesture Recognition: The detected gesture is matched with a predefined gesture library to identify the corresponding command. For example, a fist indicates stopping, an extended index and little finger indicate moving forward, an extended thumb and little finger indicate moving backward, an extended index finger indicates turning left, and an extended little finger indicates turning right. Command Generation: Based on the recognized gesture, a corresponding control command is generated and sent to the Arduino via serial communication. Vehicle Control: The Arduino receives the control command and drives the motors and actuators to perform the corresponding actions.

Safety Measures Environmental Testing: Test the vehicle's movements in a safe environment to ensure the system's stability and reliability. Hardware Inspection: Ensure all hardware components are correctly connected, and the Arduino program is correctly written to respond to received commands.

Future Prospects. Extended Applications: This system is not only applicable to vehicle control but can also be extended to other smart devices and robotic control fields. Improvement Directions: Future improvements can focus on optimizing gesture recognition algorithms to enhance the system's response speed and accuracy

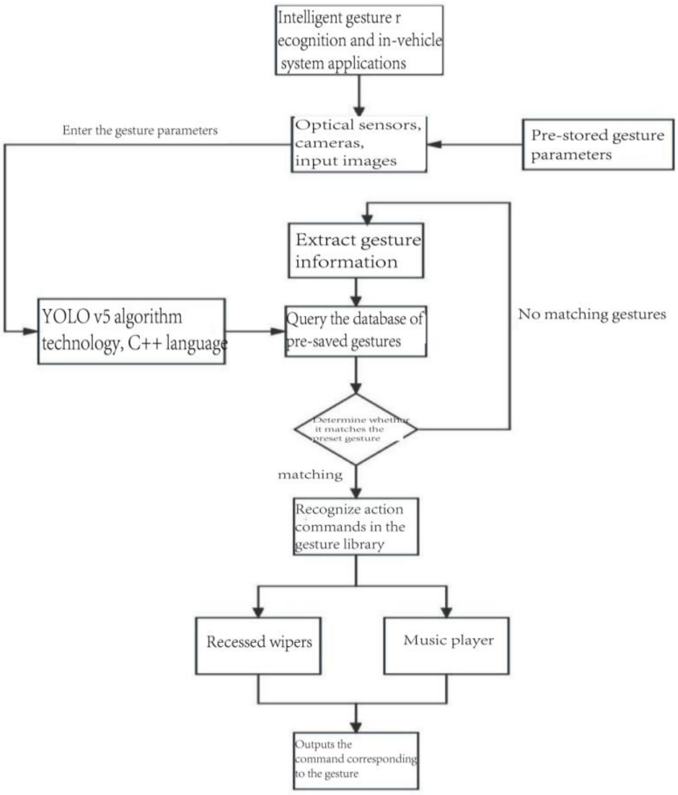

Figure 1: Flowchart of Intelligent Gesture Recognition and In-Vehicle System Based On OpenCV

2.4. Main Issues to be Addressed

2.4.1. Tracking of Gesture Actions and Subsequent Computer Data Processing

Gesture action capture is mainly achieved through optical and sensor methods. [4]Optical methods typically involve using cameras to capture images or video streams of gestures, which are then processed using computer vision techniques to identify and track hand movements. Sensor methods, on the other hand, may use devices like data gloves equipped with flexible fiber optic sensors to detect physical changes associated with gestures.

2.4.2. Design Based on QT

The design primarily leverages the robust capabilities of YOLOv5, coupled with the computational prowess of GPU training, to iteratively refine the model over 1000 epochs, starting from the yolov5l architecture. This meticulous training regimen culminates in the generation of the last. pt model file, which boasts an impressive average gesture recognition accuracy of 94%, demonstrating remarkable stability and reliability. The entire suite of libraries and technologies employed in this endeavor is firmly grounded in the versatile and widely-used Python programming language, ensuring seamless integration and extensive community support. For the upper computer interface, the design predominantly utilizes the powerful and flexible QT framework, providing a user-friendly and intuitive graphical interface for interacting with the gesture recognition system.

2.4.3. Initial Low Recognition Rate on the Upper computer

The subpar performance observed on the training set was attributable to underfitting, a condition where the model's generalization ability was excessively robust, consequently resulting in a heightened misrecognition rate. To rectify this issue, a multifaceted approach was adopted, which included augmenting the number of positive samples within the dataset, bolstering the quantity of samples pertaining to the main features, escalating the number of training iterations, and diminishing the regularization parameters. These strategic adjustments culminated in a significant enhancement, achieving a commendable gesture recognition accuracy of 94%.

2.4.4. Initial Non-Timely Response of the Lower Computer to Gesture Commands

Issues such as non-timely response or incorrect response to gesture instructions were effectively addressed through a comprehensive troubleshooting process. This process involved meticulously checking for any anomalies in TCP communication, rigorously verifying the business logic immediately after signal reception, and ensuring the seamless availability and functionality of all relevant function calls. By implementing these meticulous checks and balances, the system's reliability and responsiveness to gesture instructions were significantly enhanced, thereby providing a more robust and user-friendly experience.

2.5. Installation Environment

2.5.1. Hardware Environment

The hardware demonstration environment is an Intel(R) Core(TM) i7-8750H CPU @ 2. 20GHz 2. 21 GHz, utilizing the integrated graphics of the host rather than a dedicated graphics card for gesture recognition. This setup is designed to simulate the GPU power shortage often encountered in an embedded environment. The actual hardware environment, in contrast, is a Raspberry Pi 3B+, which provides a more realistic testbed for evaluating the system's performance under resource-constrained conditions.

2.5.2. Software Environment

The software environment is meticulously configured with a suite of dependencies that are specifically tailored to meet the stringent requirements of YOLOv5. This includes a robust Python runtime environment with version 3. 8. 8, complemented by essential libraries such as numpy 1. 18. 5 for efficient numerical computations, opencv-python 4. 1. 2 for advanced image processing capabilities, PyYAML 5. 3. 1 for handling configuration files, scipy 1. 4. 1 for scientific computing, torch 1. 7. 0 and torchvision 0. 8. 1 for deep learning and computer vision tasks, tqdm 4. 41. 0 for progress bar utilities, tensorboard 2. 4. 1 for visualization of training metrics, and seaborn 0. 11. 0 for enhanced data visualization. This comprehensive setup ensures that the system is fully equipped to run YOLOv5 with optimal performance. Additionally, the system configuration is finely tuned for the Raspberry Pi 3B+, taking into account the unique hardware constraints and capabilities of this embedded platform to ensure seamless and efficient operation.

3. System Implementation and Evaluation

3.1. First Section

The upper computer implements the gesture recognition functionality, which is fundamentally based on the advanced YOLOv5 framework. [2]As illustrated in Figure 2, the prediction training framework meticulously trained numerous iterations of the dataset. Subsequently, the training outcomes were rigorously compared and evaluated to identify and select the optimal training model, thereby ensuring the highest possible accuracy and efficiency in gesture recognition.

Figure 2: Gesture Recognition to Display English Words

3.2. Second Section

The lower computer realizes the music playback function. The music playback function is developed using Qt 5. 9. The addition of the class qfloat16, which abstracts 16-bit floating-point numbers, improves the performance of APIs using GPUs. Qt Web Socket uses external TCP sockets for convenient communication. [3]Based on the Qt Creator integrated embedded development environment, applications can be deployed across different desktop and embedded operating systems.

Data management is based on SQLite. The entire database is stored in a single file, which can be freely shared between machines with different byte orders. It is faster than popular database engines in most common operations and has a simple structure.

4. Conclusions

The system successfully applies the OpenCV library for gesture image preprocessing, feature extraction, and classification recognition, achieving high recognition accuracy and stability. In the image preprocessing stage, steps such as graying, filtering, and binarization effectively reduce noise interference and improve image quality. In the feature extraction stage, feature descriptors suitable for gesture recognition, such as HOG and LBP, are used to capture key gesture information. In the classification recognition stage, machine learning algorithms such as Support Vector Machine (SVM) or Convolutional Neural Network (CNN) are selected, and trained models are used to accurately recognize gestures. Experimental results show that the system performs well in various lighting conditions and gesture actions. The design and implementation of an intelligent gesture recognition and in-vehicle system based on OpenCV is a challenging and innovative task. Through the development and testing of the system, the feasibility and practicality of the technology have been verified, and the future development direction and market potential have been anticipated. It is believed that in the near future, this technology will play a more important role in in-vehicle systems, bringing more convenience and pleasure to people's travel.

Acknowledgements

Jilin Provincial College Students' Innovation and Entrepreneurship Plan Project in 2023 "Design and Implementation of an Intelligent Gesture Recognition and In-Vehicle System Based on OpenCV" (Project Number: 202310201026)

References

[1] Lu Haihong, "Computer Measurement and Control," Xi'an University of Posts and Telecommunications, Institute of Internet of Things and Integration of Two Industries, 2015.

[2] Zhu Wenwen, Ye Xining. "The Gesture Recognition Algorithm Based on a Convolutional Neural Network". Journal of East China University of Science and Technology (Natural Science Edition), 2018, (2).

[3] Miao Qiguang, Li Yinan, Liu Xiangzeng, Liu Ruyi, "Gesture Recognition Technology Theory and Application," Xidian University Press, 2022.

[4] Shen Yating, "Software Guide," Nanjing University of Science and Technology, Zijin College, Computer Science and Technology Department, 2019.