electronics

MDPI

Article

Impact of In-Air Gestures on In-Car Task's Diver Distraction

Chengyong Cui [1], Guojiang Shen [1], Yu Wang [1], Yile Xu [2], Hao Du [3], Wenyi Zhang [1] and Xiangjie Kong [1,*]

[1] College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China
[2] Computer Science and Interdisciplinary Studies, College of William and Mary, Williamsburg, VA 23186, USA
[3] Key Laboratory of Public Security Information Application Based on Big-Data Architecture, Ministry of Public Security, Zhejiang Police College, Hangzhou 310053, China
* Correspondence: xjkong@ieee.org

Abstract: As in-vehicle information systems (IVIS) grow increasingly complex, the demand for innovative artificial intelligence-based interaction methods that enhance cybersecurity becomes more crucial. In-air gestures offer a promising solution due to their intuitiveness and individual uniqueness, potentially improving security in human–computer interactions. However, the impact of in-air gestures on driver distraction during in-vehicle tasks and the scarcity of skeleton-based in-air gesture recognition methods in IVIS remain largely unexplored. To address these challenges, we developed a skeleton-based framework specifically tailored for IVIS that recognizes in-air gestures, classifying them as static or dynamic. Our gesture model, tested on the large-scale AUTSL dataset, demonstrates accuracy comparable to state-of-the-art methods and increased efficiency on mobile devices. In comparative experiments between in-air gestures and touch interactions within a driving simulation environment, we established an evaluation system to assess the driver's attention level during driving. Our findings indicate that in-air gestures provide a more efficient and less distracting interaction solution for IVIS in multi-goal driving environments, significantly improving driving performance by 65%. The proposed framework can serve as a valuable tool for designing future in-air gesture-based interfaces for IVIS, contributing to enhanced cybersecurity.

Keywords: artificial intelligence technology; cyber security; interaction; gesture recognition; driving simulator

check for updates

Citation: Cui, C.; Shen, G.; Wang, Y.; Xu, Y.; Du, H.; Zhang, W.; Kong, X. Impact of In-Air Gestures on In-Car Task's Diver Distraction. *Electronics* **2023**, *12*, 1626. https://doi.org/10.3390/electronics12071626

Academic Editor: Juan M. Corchado

Received: 28 February 2023
Revised: 24 March 2023
Accepted: 28 March 2023
Published: 30 March 2023

1. Introduction

In recent years, the complexity of in-vehicle information systems (IVIS) has increased, and cybersecurity in the interaction has become a hot topic [1]. The challenge in designing such a system is to balance the need for a pleasant and personalized interface with the need to ensure drivers' security and reduce driver distraction [2,3].

The use of IVIS while driving creates a multi-goal environment, where the primary objective is to drive safely while the secondary objective is to use the IVIS [2]. However, if the driver's cognitive resources are primarily occupied by the IVIS screen, it can pose risks to personal security. Therefore, an effective IVIS should help users allocate cognitive resources properly, address usage problems in a multi-goal environment, and ultimately improve both driving safety and interaction security [1].

While touch, click, and other interaction methods are widely used on IVIS [3,4], recent advancements in artificial intelligence technology have led to the emergence of new interaction methods, such as voice input and in-air gesture recognition [5]. These methods that use biometric information can greatly improve cybersecurity in interaction. In particular, in-air gesture recognition allows drivers to perform customized gestures within the recognition area of a sensor, such as a camera or a sensor, which can then provide personalized feedback through the IVIS system [4]. As users do not need to look at the screen when making gestures, we believe that in-air gesture recognition has the potential to reduce driver distraction and improve driving safety compared to touch interactions.

Additionally, here are some influencing factors for gesture recognition, which mainly include the following aspects:

1. Lighting conditions: Gesture recognition systems are highly sensitive to lighting, and strong light or shadows may interfere with gesture recognition [6].
2. Background noise: Noise or interfering objects in the background may also cause disruptions to gesture recognition [7,8].
3. Camera position and viewing angle: The position and angle of the camera can affect the visual features of gestures, thereby influencing the accuracy of gesture recognition [6,9].
4. Gesture diversity and variations: Different shapes and motions of gestures may lead to changes in visual features, thus affecting the accuracy of gesture recognition [7].
5. Quality and diversity of datasets: The performance of gesture recognition systems largely depends on the quality and diversity of training data. If the training dataset is insufficient or not representative, the accuracy of gesture recognition may be affected [7]. It is also mentioned that there are currently few effective datasets available for training gesture recognition models, and even fewer datasets with accurate annotations for hand actions [7,10].

However, at present, there is limited research on the recognition accuracy of gesture recognition methods based on visual input (using only RGB images) in an in-vehicle environment. Moreover, there is little research on whether such interaction methods can reduce driver distraction during interactions with in-vehicle IVIS systems. Additionally, there is a lack of research on classification methods suitable for use with the Mediapipe framework.

To address these gaps, our paper makes the following contributions:

- We develop an evaluation system to measure the level of driver distraction caused by two different interaction methods: gesture and touch. Specifically, in the assessment of the driver's primary task, calculate the driving scores under different interaction methods.
- We proposed a gesture recognition framework based on Mediapipe suitable for Android mobile devices and tested it on the AUSTL dataset alongside two state-of-the-art gesture recognition methods.
- We design a custom IVIS system interface and build a simulation environment to mimic driving scenarios. At the same time, use user experiments to select the most suitable gestures for the system's gesture interaction.

By improving our understanding of the potential benefits of in-air gesture recognition in the in-vehicle environment, this research can help inform the development of more effective and safer IVIS interfaces for drivers.

2. Related Work

This section discusses three aspects related to this paper: the design of IVIS interaction interfaces, the in-air gesture recognition methods, and factors causing driver distraction.

2.1. Design of IVIS Interaction Interfaces

As the number of IVIS functions increases, the design requirements for the interface have become more challenging. The common method is to use user studies to ensure that the interface is easy to use and welcomed by drivers, while also ensuring driving safety [1,2]. For an interface that uses in-air gestures, the design should ensure that all gestures are commonly used and easy to operate for most people. Researchers have used user studies to determine the most easy-to-use flip gestures in dual-screen mobile phones [11]. User experiments are also used to determine the best design of interaction interfaces [1,4,12,13].

Additionally, when gesture interaction is impaired due to noise or environmental interference, audiovisual speech recognition can be employed for interaction [14–16]. One approach for sentence-based speech recognition in IVIS systems is discussed in [15]. Tests on the famous LRW lip reading dataset have demonstrated that high-performance models for recognizing drivers' lip movements can be obtained even when only using the video modality for training. Moreover, a systematic study on improving word classification accuracy has been conducted in another paper [14]. The authors found that temporal masking (TM) is the most crucial enhancement method, followed by mix-up, and that densely connected temporal convolutional networks (DC-TCN) are the best temporal models for isolated word lip reading. By combining these methods, the resulting classification accuracy reaches 0.934. With further training, the recognition accuracy can be improved to 0.941. Lastly, multi-head visual-auditory memory (MVM) can address two challenges in lip reading [16]. MVM consists of a multi-head key memory and a value memory, which distinguishes homophones. It extracts audio representations solely from visual inputs, thereby supplementing the lip reading model's visual information. The final experimental results validate the effectiveness of this method in differentiating homophones.

2.2. In-Air Gesture Recognition Methods

Firstly, with the growing interest in the field of gesture recognition and sign language recognition, there are now many excellent datasets available for use. Based on their time and composition, we summarize the commonly used datasets as Table 1.

Table 1. Collection of datasets.

Datasets	Time	Components
InterHand2.6M [17]	2020	2.6 million frames of hand keypoint annotations, including a total of 1068 hand models.
AUTSL [18]	2020	A Turkish sign language dataset with 226 signs, comprising 38,336 videos.
WLASL [19]	2019	An American sign language dataset with over 2000 gestures, containing around 47,000 videos.
ChaLearn Pose [20]	2013	An Italian gesture collection with information on 20 gestures and other body parts.
LSA64 [21]	2016	An Argentine sign language collection with 64 gestures, containing 3200 videos.
MS-ASL [22]	2018	A collection of more than 25,000 real-life American sign language videos.
Cambridge Hand Gesture datasets [23,24]	2007	A dataset with 9 gestures, containing 900 videos of both static and dynamic gestures.
Northwestern University Hand Gesture datasets [23,25]	2009	A dataset with 10 gestures, containing 1050 videos under different background conditions.

What is more, in-air gesture recognition methods can be hardware-based or image processing-based [26,27]. Hardware-based devices, such as Leap Motion sensors and Microsoft Kinect sensors, capture more features than images, making them more resistant to interference. However, the cost of these devices is high, making large-scale deployment challenging. Image processing-based methods are more accessible and have lower implementation costs. The most common method is to segment different parts of the palm using color block marks or a skeleton-based approach, as demonstrated in Google's Mediapipe framework [27–29]. Skeleton-based recognition methods have a higher recognition accuracy and can be used with only an RGB camera, making them suitable for large-scale deployment in life scenarios.

Meanwhile, with the development of LSTM and Attention mechanisms, some state-of-the-art gesture recognition methods have also adopted these techniques. The LSTF+LSTM recognition method consists of two main modules for gesture recognition [8]. Firstly, the feature extraction module uses a 64-dimensional BiLSTM layer to obtain various hand features. Secondly, the feature recognition and classification module inputs the STF features extracted in the previous step into the Attention layer, followed by a 32-dimensional

BiLSTM layer and an FCNN layer to output classification results. The final recognition accuracy reached 0.9856. The SSTCN method for recognizing gesture skeleton features [7] involves using a GCN to establish the whole-body keypoint SL-GCN module and obtain the pose network for the entire body. The Skeleton Aware Multi-modal SLR framework is then proposed to improve recognition accuracy in multi-modal settings (RGB and RGB-D). The final recognition accuracy reached 0.9853. The issue of insufficiently large datasets for gesture recognition has also been addressed [30]. The authors use a large-scale model trained on general datasets and fine-tune it for specific downstream tasks. Using the SAM-SLR framework and fine-tuning, the authors achieved a recognition accuracy of 0.9572 on the WLASL and AUTSL validation subsets. The Video Transformer Network (VTN) is used for gesture recognition [31]. The authors use a transformer to recognize gestures with only RGB input data. Deep CNNs are used for spatial information modeling, while self-attention is employed for temporal information modeling. The final recognition accuracy reached 0.9292.

In the gesture recognition methods used above, a common issue is the recovery of corrupted image frames. There are several main approaches to image restoration currently available. One approach involves the use of a novel rank minimization problem method, referred to as the Rank Residual Constraint (RRC) model [32]. This method gradually approximates the underlying low-rank matrix by minimizing the rank residual, and the experimental results outperform many state-of-the-art schemes. Another approach involves considering both internal and external non-local self-similarity (SNSS) priors simultaneously to provide complementary information [33]. Based on this, the authors propose an alternating minimization method with an adaptive parameter adjustment strategy to address the SNSS-based image restoration problem. The final experiments demonstrate that the proposed SNSS produces superior results in terms of objective and quality measurements compared to many popular or state-of-the-art methods. Lastly, there is a new sparse representation model called Joint Block-Group-Based Sparse Representation (JPG-SR) [34]. This model is based on the Alternating Direction Method of Multipliers (ADMM) framework and utilizes an iterative algorithm.

2.3. Factors Causing Driver Distraction

Research on driver distraction in multi-goal environments focuses on various factors. The effect of IVIS screen size on interaction efficiency and driver interference was studied in [35], where subtasks had a significant effect on driving attention. The design of the IVIS system should divide tasks into smaller ones to reduce the time drivers spend on the interface during each operation, thus improving driving concentration. IVIS operation during driving is a multi-goal environment, where the primary mission is to drive, and the secondary mission is to use IVIS [36]. Research has also focused on how the auditory interface affects drivers and how to improve it [35,37,38].

Comparative experiments have been conducted on the auditory and visual interfaces, where the former had less driver interference. However, the accuracy of speech recognition in noisy environments, such as the sound of the engine during driving, is a challenge. Moreover, the current speech recognition technology cannot cope with long text instructions [35].

3. Methodology

3.1. In-Air Gesture Recognition Framework

Based on the related work, we decided to utilize a skeleton-based approach for in-air gesture recognition. To achieve real-time acquisition of the palm skeleton, we utilized a framework consisting of the BlazePalm Detector and HandLandmark Model, as described in [27,29].

When a video stream is input for the first time or the palm appears in the input video stream for the first time, the BlazePalm Detector model is called to determine the palm's position in the video frame. The framework then transmits the cut palm image to the

Hand Landmark Model to determine the 21 key points of the palm in 3D coordinates. This approach allows us to obtain the necessary key point information of the palm.

Next, we extract relevant features of the in-air gesture from the key point coordinates to realize recognition and classification of the gesture. Based on their characteristics, we classify in-air gestures into two types: static and dynamic. This is because dynamic gestures and static gestures have three distinct features [8]: 1. preparatory gesture, 2. context-independent gesture motion, and 3. retraction action. The preparatory gesture represents the initial action state of the dynamic gesture, while the context-independent gesture motion serves as the core feature that distinguishes each gesture from others. The retraction action indicates the preparation for the next gesture. Moreover, the retraction action for some dynamic gestures is very similar to the preparatory gesture, which causes difficulties in recognizing dynamic gestures [9,10].

For static gestures, we rely on the finger extension of the palm and geometric information to make judgments. We use Equation (1) to judge finger extension. To determine the orientation of the palm, we divided it into four directions: up, down, left, and right. Using key point 0 as the coordinate origin, we calculate the angle between the palm direction vector and the horizontal direction vector to determine the orientation of the palm. In Figure 1, the vector from the origin pointing to the 18th coordinate point is used as the direction vector of the palm. As shown in Figure 2, the orientation of the palm is determined based on the rotation angle of the palm.

$$\begin{cases} (y_i - y_{i+1})(y_{i+1} - y_{i+2})(y_{i+2} - y_{i+3}) > 0 \\ (x_i - x_{i+1})(x_{i+1} - x_{i+2})(x_{i+2} - x_{i+3}) > 0 \end{cases} \tag{1}$$

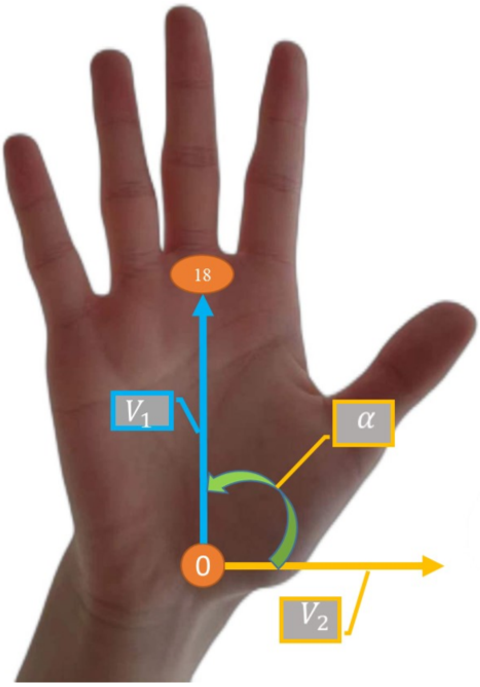


Figure 1. Vectors and angles on the palm. The orientation of the palm is determined based on two vectors and the rotation angle. V_1 represents the direction vector of the palm, and V_2 represents the horizontal vector of the palm. α is the rotation angle of the palm.

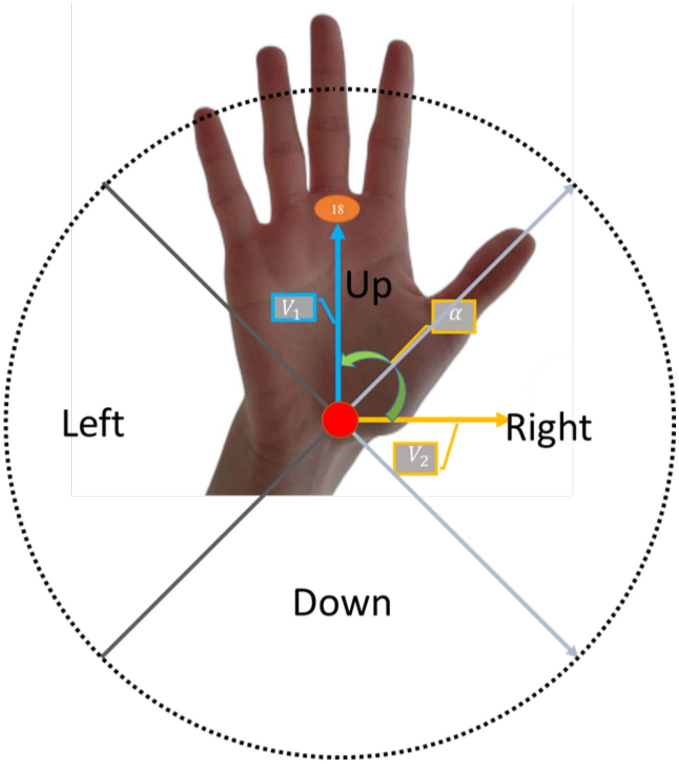


Figure 2. Determining the palm orientation. The contents marked with V_1, V_2, and α have the same meaning as in Figure 1.

For dynamic gestures, the data recorded include not only the extension and geometric information of the palm, but also the timing information of the action, which forms a time series of key point coordinates.

To extract the start and end states of a dynamic gesture sequence, we can use the static gesture recognition method for recognition. When the gesture recognition framework recognizes that a gesture is the start state of a dynamic gesture, it automatically records the sequence of gesture actions until the framework recognizes the end state of the dynamic gesture. The system then stops the recording of gesture key points and starts the calculation. Therefore, we can determine the type of dynamic gesture by collecting the key point series of dynamic gestures, and then use related methods to calculate the similarity for classification.

We studied three common classification methods suitable for the MediaPipe framework [18,23,24]: (1) the BP neural network approach, with the structure of the neural network model shown in Table 2; (2) the LSTM network, with the structure of the LSTM network model shown in Table 3; and (3) by calculating the distance between the sampling sequence and the template sequence. The commonly used method for calculating the time series distance is dynamic time warping (DTW). DTW finds the matching points corresponding to each key point of the sampling sequence and the template sequence through the dynamic programming algorithm, and then obtains the distance between the two time series by accumulating the distance between these key points.

Table 2. Architecture of BP network.

Layers	Input	Output
Input	Array	1 × 40
Hidden	1 × 40	1 × 24
Hidden	1 × 24	1 × 10
Output	1 × 10	1 × 2

Table 3. Architecture of LSTM network.

Layers	Input	Output
Input	Array	1×40
Reshape	1×40	$(20, 2)$
LSTM	1×20	1×20
Hidden	1×20	1×10
Output	1×10	1×4

To determine the appropriate classification method for the Mediapipe framework, we first select small-scale datasets, such as the Cambridge [23] and the Northwestern University Hand Gesture dataset [39] for testing. The recognition performance of different classification methods is shown in Table 4.

Table 4. Comparison between three different classification methods.

Data Set	BP	LSTM	DTW
Cambridge	92.37 ± 1.67%	90.33 ± 2.78%	89.33 ± 2.88%
Northwestern	80.13 ± 1.89%	80.25 ± 1.86%	81.23 ± 1.49%

Based on the results of testing with small-scale datasets, we found that the DTW and BP neural network classification methods, combined with the MediaPipe framework, achieve higher recognition accuracy. In related work, we introduced many state-of-the-art recognition methods, among which SAM-SLR [7] and VTN-PF [31] achieve relatively high recognition accuracy. To analyze the gap between the recognition method we proposed based on the MediaPipe framework and the state-of-the-art methods, we validate using the large-scale AUTSL dataset [18].

Considering that the core of the paper is to analyze the distraction of gesture recognition methods on drivers, when selecting a gesture recognition method, in addition to using recognition accuracy as a measurement, we also need to consider the performance of the recognition method on resource-limited mobile devices. A recognition method, even with high recognition accuracy, if it is too computationally complex or consumes too many resources, may result in long recognition times or device lag and program crashes on mobile devices. This would significantly affect the interaction between the driver and the IVIS system in the in-car environment, distracting the driver's attention when using gesture interaction and affecting the effectiveness of the subsequent comparative experiments conducted in the paper.

Therefore, we refer to the evaluation method for neural networks on edge devices in [40], and while testing the accuracy of different recognition methods, we also pay attention to the GPU occupancy rate of the recognition methods. The GPU occupancy rate can be calculated through the Profiler in Android Studio. Finally, with recognition accuracy as the horizontal axis and GPU occupancy rate as the vertical axis, the positions of several different recognition methods on evaluation indicators are shown in the Figure 3 and Table 5.

Table 5. Testing results of different gesture recognition methods on the AUTSL dataset.

Recognition Method	Accuracy	GPU Occupancy Rate
SAM-SLR	**97.31**%	89%
VTN-PF	91.37%	83%
Mediapipe+CNN	92.68%	55%
Mediapipe+BP	88.37%	51%
Mediapipe+DTW	80.65%	**49.5**%

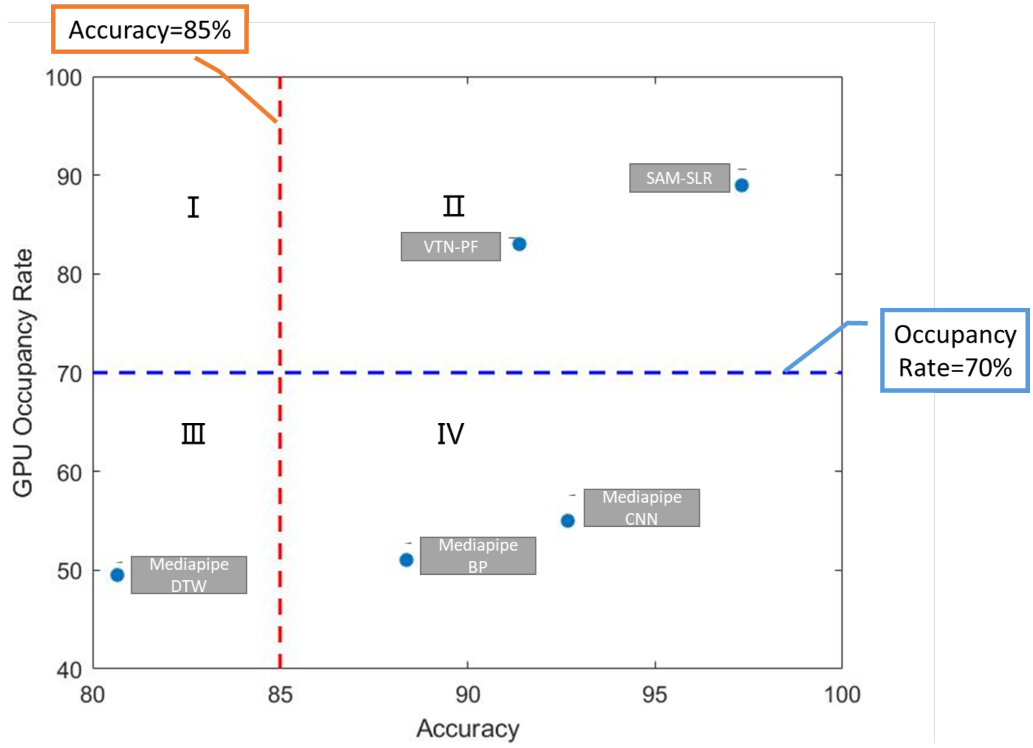


Figure 3. AUTSL large-scale dataset test results on Android mobile devices. Each point in the figure represents a recognition method. Coordinate values are percentages.

In the figure, we divide the area into four parts according to a GPU occupancy rate of 70% and a recognition accuracy rate of 85%. The recognition methods in Zone IV are the closest to the ideal situation, with relatively high recognition accuracy and low GPU occupancy rates. In contrast, the recognition methods in Zones I and II are not suitable for use in in-vehicle IVIS systems. These methods have high GPU occupancy rates (above 70%), which can easily cause unnecessary device lag and crashes. We find that the recognition accuracy of the two state-of-the-art methods being compared is very high, which may be because both methods simultaneously recognize the movements of other body parts (such as lips) in coordination with the hands during sign language. However, the Mediapipe framework using CNN as a classifier also has a close recognition efficiency. In terms of GPU occupancy rate, the two state-of-the-art methods consume too much GPU resources, occupying nearly 30% more than the methods using the Mediapipe framework. One reason for this result is that the Mediapipe framework optimizes data packet communication and transmission for the Android system [27], while the two state-of-the-art methods currently being compared do not have optimizations for mobile devices. Therefore, we choose the Mediapipe framework in Zone IV.

Moreover, the test results using the AUTSL large-scale dataset are significantly different from the previous test results using small-scale datasets. CNN's classification performance on large-scale datasets is significantly better than that of BP and DTW methods. This is because the input data dimensions of BP and DTW classification methods are limited, and the complexity of the calculations increases with the number of classifications. Especially as a traditional classification method, the recognition accuracy of the DTW method heavily depends on the template sequence, and it has a low tolerance for input noise.

However, considering that in the following comparative experiments in the paper, the number of gestures is limited and predetermined, we still refer to the test results of the first small-scale dataset in choosing the classification method. In summary, taking into account the recognition accuracy and resource occupancy of the recognition methods, we choose the Mediapipe-based recognition framework and establish a gesture recognition framework represented by the results obtained in Table 4, which will serve as the gesture recognition method for the IVIS system we will build next.

In conclusion, we have developed a skeleton-based method for classifying in-air gestures and presented a comprehensive framework for the gesture recognition system, as shown in Figure 4. Our framework combines the BlazePalm Detector and HandLandmark Model for real-time acquisition of palm skeleton, and the static and dynamic gesture recognition methods for feature extraction and classification.

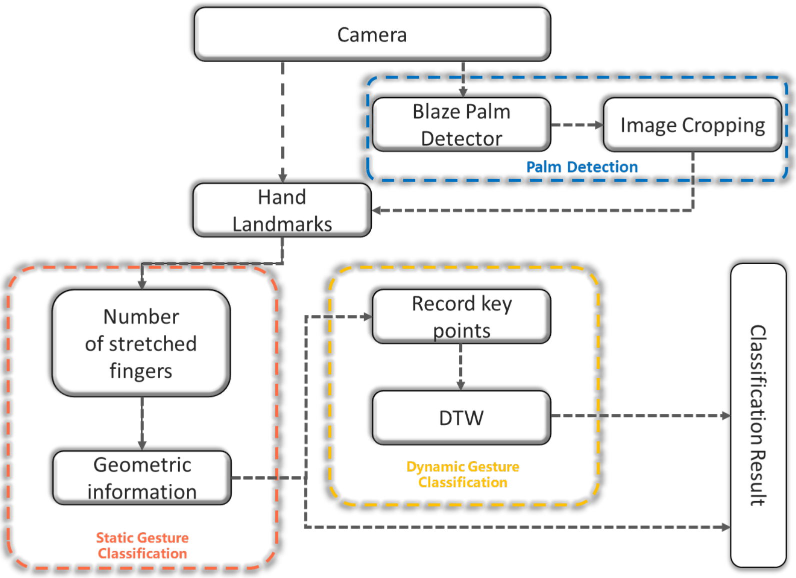


Figure 4. Gesture recognition framework. The Palm Detection module is only used when the first frame of the input camera image and the first palm appears in the image frame.

3.2. User Study

Our research aims to identify the most appropriate gestures for in-vehicle interaction and investigate whether the in-air gesture interaction method can minimize driver distractions compared to the traditional touch interaction method. To achieve this, we designed a simulation environment that mimics the driving environment and conducted two user studies. The following sections detail the process of creating the simulation environment and conducting the user studies.

3.3. System Design

3.3.1. System Function

Before designing the system interaction interface, we conducted user research to select commonly used driver operations in the IVIS system as task content. We collected opinions from research participants with driving qualifications through group discussions and questionnaires. The final set of functions that our IVIS system includes are shown in Figure 5 and can be grouped into three main sections:

1. Music control: This section includes music playback, pausing, volume adjustment, and song switching.
2. Map control: This section includes map movement, zooming, and positioning.
3. Call answering: This section includes answering and hanging up mobile phone calls.

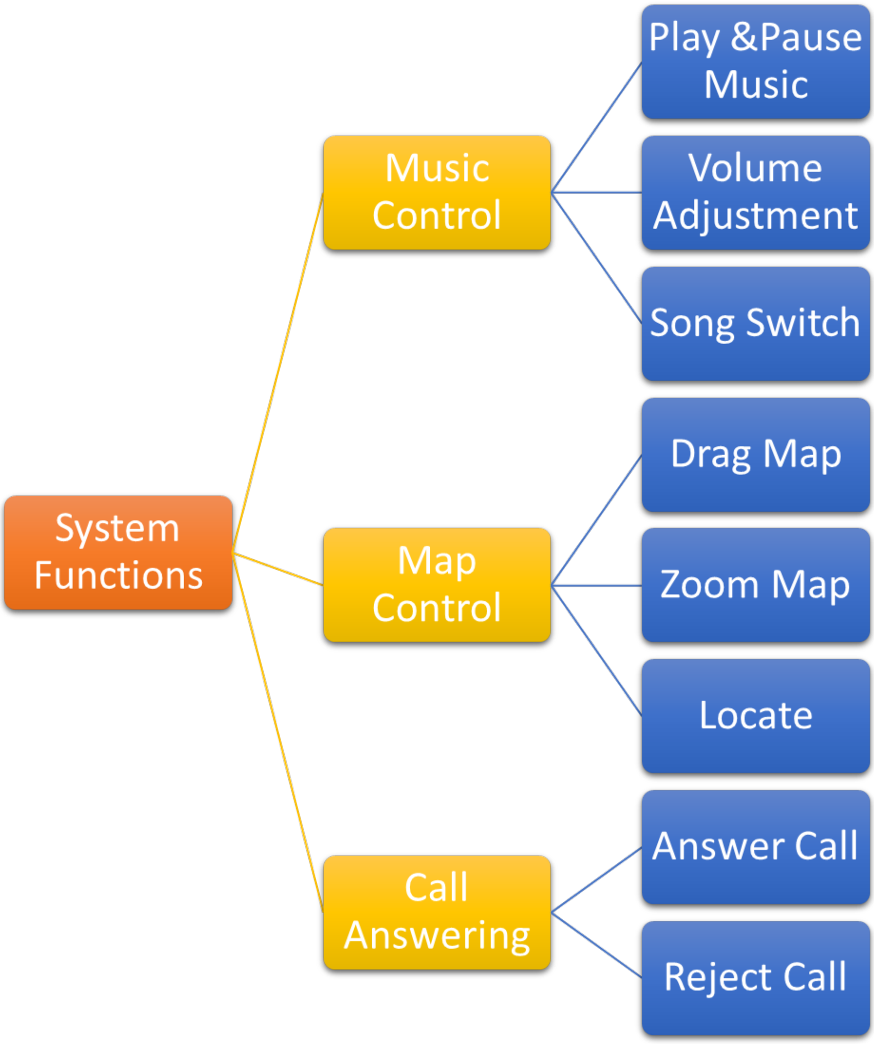


Figure 5. System function.

3.3.2. Interface Design

In [36], it is suggested that the touch interface designed for the driver must have eye-catching functions. Since our system's functions can be categorized into three main parts, we opted to use a multi-level menu design for the touch interface. However, we found that the incoming call control only has two functions, so including too many branches in the menu could lead to operating errors and reduced efficiency. Thus, our final touch interface design consists of three main interfaces: the main menu, music control interface, and map control interface. In the main menu (refer to Figure 6), the menu bar options lead to two secondary menus. In the music control interface (refer to Figure 7), we added a module to display music and volume information. This shows the song information and volume status that the system is currently playing. For the map control interface (refer to Figure 8), we added a feature to display the current location on the map. The map and location information services use the API interface provided by Gaode Maps. To enhance the interface's intuitiveness, we chose icons for each manipulation task that will appear in a pop-up window when the system executes the task. The location of the pop-up is illustrated in Figure 7.

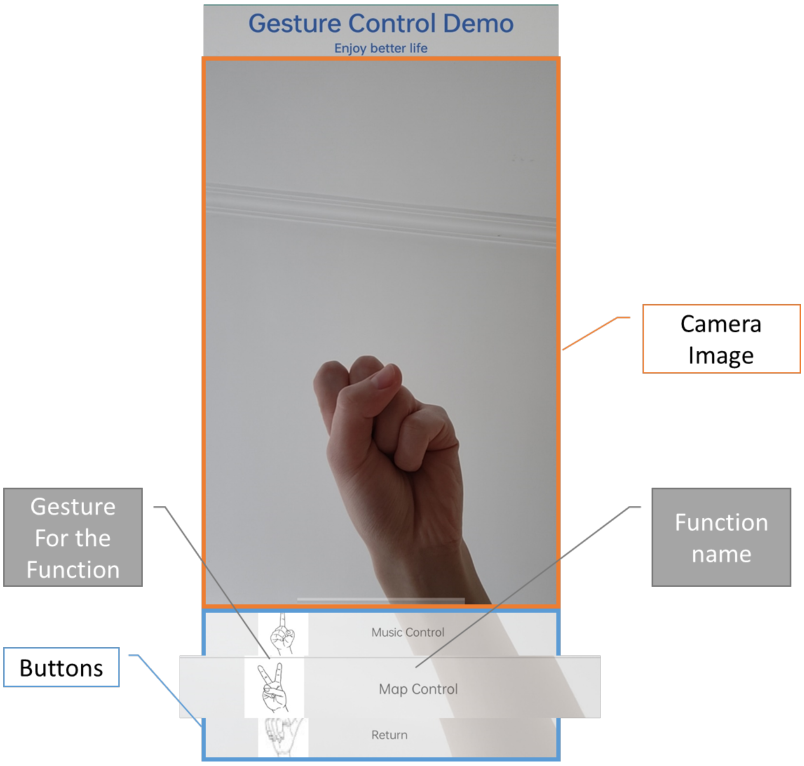


Figure 6. Main menu. The location of the Camera Image and Buttons is shown in the picture. Each button adds a gesture photo and function name.

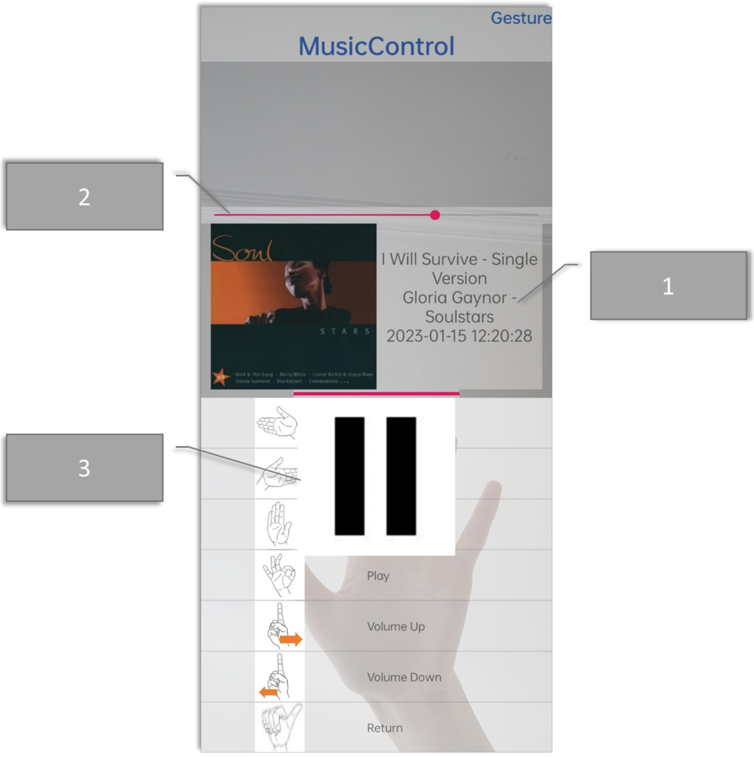


Figure 7. Music control interface. No. 1 indicates the detailed information and cover of the music being played. No. 2 represents the volume display. No. 3 indicates the pop-up window when the operation is completed. Currently displayed is the pop-up window for the completion of the Music Pause task.

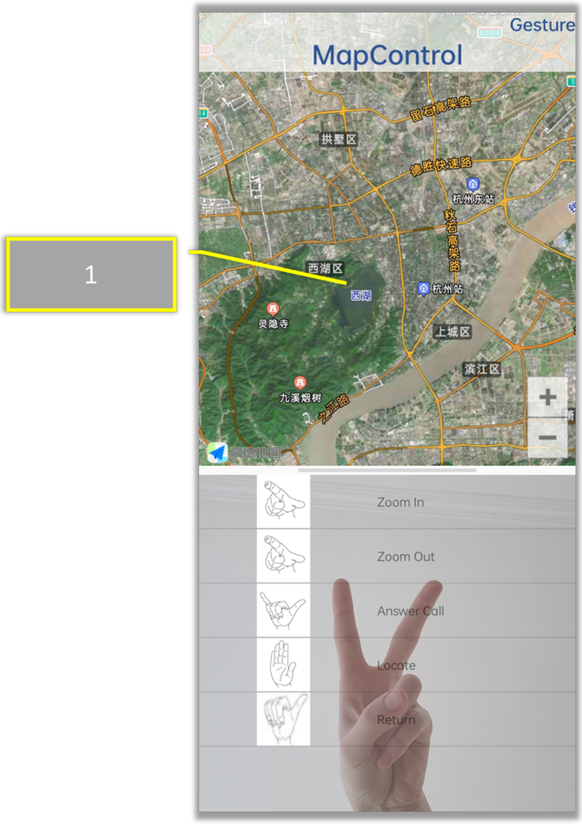


Figure 8. Map control interface. No.1 represents the position displayed on the map.

To enable the driver to perform in-vehicle tasks without looking at the interface, we designed an in-air gesture interaction interface that assumes the driver is familiar with the gestures of various system operations, and the system's gesture recognition accuracy is high. However, during our research, we found that even if the driver is familiar with our IVIS and the gestures, they still need to check the screen, especially to view the image display of the smartphone's front camera to ensure their gestures are recognized within the camera's recognition area. Therefore, we added the image display part of the front camera to all three interfaces of the touch interface we designed previously.

To further reduce the system's interference with the driver, we included a visual feedback method that displays the task icon in a pop-up window upon completion of the operation. Additionally, we incorporated acoustic feedback for each operation task in the form of recorded prompts, which are played after the correct performance of the operation task, to minimize the number of times the driver needs to look at the screen. Acoustic feedback has been demonstrated to be an effective method in reducing the driver's distraction in previous work [41].

3.4. Experiment Design

In this section, we will describe the various variables that were set corresponding to the environment in the two user studies. For the study focused on determining the most suitable in-air gesture for system interaction, the variables included the position of the gesture recognition camera and the gestures used in our IVIS. In the study comparing in-air gesture and traditional touch interaction, the experimental variable was mainly the mode of interaction. Therefore, in these two user studies, cameras were installed in the center panel and the rear console for comparative experiments, and the interface was designed for the two interaction methods.

3.4.1. Experimental Environment Preparation

Participants

We recruited 20 participants for this study, all of whom were eligible drivers with an average age of 23 and an average driving experience of 4 years. Among the participants, 4 were left-handed (20%) and 8 were female (40%). We also recruited an additional control group of 4 participants (2 males and 2 females) to evaluate their driving performance in the simulated environment.

Driving Simulation Environment

To simulate the driving environment, we constructed a driving simulation environment in the laboratory using hardware devices and displays [37,42]. Our driving simulator included a steering wheel, gears, brakes, accelerator, and clutch. A 55-inch LCD screen was used to display the simulated road environment (see Figure 9), which was obtained from a driving simulator and featured an urban arterial road with less traffic and straight roads. The speed was set to a constant 60 Km/h. We used a 6.5-inch Android smartphone as the IVIS system in the simulated environment, placed on the right side of the steering wheel and in front of the AT change lever to simulate the positions of the rear console and the center panel in a car (see Figure 9).

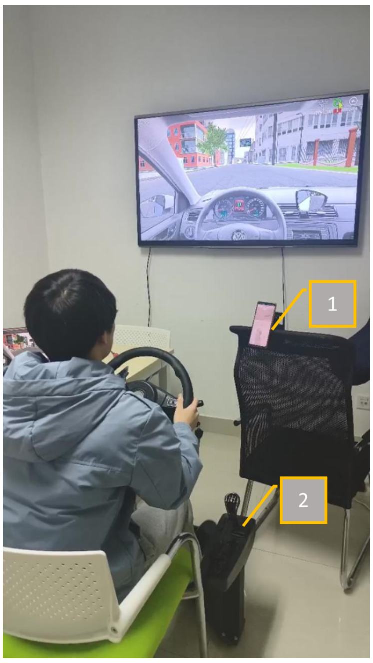


Figure 9. Camera location. No. 1 represents the simulated center panel position in the simulation environment; No. 2 represents the simulated rear console position in the simulation environment.

3.4.2. Experiment 1: Find the Most Appropriate Gesture

Experiment Design

In Experiment 1, we aimed to determine the most suitable gestures and camera positions for in-vehicle interaction, while also considering the potential influence of handedness on interaction efficiency. Since the camera's limited recognition range often requires the driver to control the steering wheel with their left hand and make gestures with their right hand, the dominant hand may affect interaction efficiency. We used a combination of objective experimental data and subjective user feedback to address these issues.

Determination of gesture collection. One of the key tasks in Experiment 1 was to identify the most appropriate in-air gestures for the driver to use. To be effective, the selected gestures should be common in daily life and easy to execute within the limited space of the car's interior. By gathering input from our driver group and leveraging our personal experience, we identified a collection of 30 gestures that meet these requirements and are well-suited for the IVIS system.

Camera positioning. Due to cost, car circuit design, and privacy concerns, cameras in cars are typically installed on the center panel or rear console [12] (see Figure 9). Therefore, this experiment will compare the interaction efficiency of these two positions to identify the best camera positioning.

Interface design. Experiment 1 determined the most appropriate gesture for the driver, so font and icon size were not a concern for the interface. Therefore, we designed a new interface that solely displays the front camera view of the smartphone. When the driver performs each gesture successfully, the program displays a pop-up window and plays a sound. The app also records the time taken for each gesture from when the experimenter issues the command until the test subject completes it, automatically collecting experimental data.

Subjective evaluation. Since Experiment 1 is a user experiment, we collected subjective evaluations from each test subject for each gesture after the experiment. To effectively measure subjective evaluations, we divided them into five levels ranging from dislike to like, with each level further divided into 20 points, totaling 100 points. We designed these evaluations in the form of questionnaires and asked test subjects to fill them out.

Procedure

We conducted a user experiment to determine the best gestures for our IVIS system. The experiment was carried out in five steps:

1. The experimenter proposed a preliminary set of gestures based on driver suggestions and personal experience.
2. The experimenter added all the gestures in the collection to the application designed in the experiment preparation.
3. Each test subject was invited to complete experimental tasks on a driving simulator. The tasks required the test subjects to make gestures corresponding to the number given by the experimenter. However, prior to the experiment, each test subject underwent a warm-up drive and familiarized themselves with the number corresponding to each gesture and the action of the gesture. The application recorded the time taken for each test subject to perform each gesture correctly during the experiment.
4. After completing the test, each test subject completed a questionnaire to indicate their preference for different gestures.
5. The experimenters analyzed both objective and subjective indicators and selected the gestures with the highest scores.

3.4.3. Experiment 2: Comparative Experiments

Experiment Design

In Experiment 2, in order to explore the influence of in-air gesture interaction and traditional touch interaction on driver’s attention, we designed a comparative experiment. In the comparative experiment, the previously designed visual interface and gesture interface were used to complete the same operation tasks.

Updates to the gesture interface. In Experiment 2, we conducted a comparative study to investigate the impact of in-air gesture interaction and traditional touch interaction on driver attention. The study involved using the previously designed visual interface and gesture interface to complete the same set of operation tasks.

Primary mission and indicators. In the multi-task scenario of driving a car, the primary mission is the driver’s focus on the driving situation. To measure the driver’s concentration on driving, we use indicators that reflect primary task completion. These indicators include driving penalty points for violations and metrics calculated by the simulation software for the driving situation. Driving penalty points are assigned for various violations, as shown in Table 6.

The objective indicators for the driving situation include the Mean Deviation (MDEV) of the lane change path and the Lane Change Initiation (LCI) detected by the simulation software [43]. MDEV indicates the driver's ability to maintain the vehicle on the desired path and provides an objective measure of their perception, maneuvering quality, and lane-keeping ability. LCI is a measure that instructs the driver to react to a sign and initiate a lane change. By combining these evaluation indicators, we can evaluate driving violations and safety.

Table 6. Penalty points for driving violations.

Violation	Penalty
Speed violation	−50
Below the speed limit	−20
Car stalled	−50
Pressure line	−20
Crash	−100
Changing lanes illegally	−20
Below the speed limit	−20

Secondary mission. The secondary mission in the multi-target scenario of driving a car is to complete the driver's tasks on the IVIS system. We evaluate the efficiency of the interaction methods by measuring the time taken by the driver to complete an operation task from receiving the experimenter's instruction. A longer completion time indicates that the current interaction method is less efficient, requiring more attention from the driver.

Subjective evaluation. In addition to the objective indicators, we also consider the driver's subjective evaluation of the gesture and touch interaction methods. We use two evaluation criteria for the driver's subjective evaluation. The first criterion is the driver's assessment of the workload of the interaction method, which we measure using the NASA TLX stress load test [12,44]. The test assesses workload on a 21-point scale, with higher scores indicating higher workload. We use the stress test questionnaire to collect the driver's subjective feelings about the attention required by the interaction method. The second criterion is the driver's preference for a certain interaction method. We designed a questionnaire to evaluate the interface, in which the driver rates their satisfaction on a 100-point scale divided into five levels, ranging from very dissatisfied to very satisfied. We combined the driver's stress load rating and preference to obtain a subjective evaluation of the interaction modality.

Procedure

1. Incorporate the gesture set identified in Experiment 1 into the application to be used in the experiment.
2. Warm-up driving: Prior to the actual experiment, participants must drive in the simulation environment to familiarize themselves with the driving simulator, the software's simulated road conditions, and all system functions. They must also be able to use touch and gestures to complete system tasks.
3. Participants begin the interface test, using both touch and gesture interaction methods to manipulate all tasks in the IVIS system. The order in which each interaction method is used is randomized, and a 15-minute break is required between experiments using the two interaction methods to eliminate the potential influence of the interaction method order on experimental results. To ensure driving safety, participants are required to hold the steering wheel with their left hand and control the task with their right hand when performing manipulation tasks.
4. Record the driving violations and task completion times for each participant during the test.

5. After completing the test, collect subjective evaluations from participants. Participants will complete a NASA TLX stress load test and a questionnaire on their interaction preferences.

4. Result and Interpretation

4.1. Experiment 1

4.1.1. Camera Positioning

In Experiment 1, we evaluated the camera's placement in two positions, namely the center panel and the rear console. To select the optimal location for the camera, we considered both objective data and subjective feedback from the drivers.

Based on the experiment results, we found that the average time for completing all 30 gestures, represented as t_a, was 95.3 s when the camera was positioned in the rear console, and 97.6 s when it was placed in the center panel. Additionally, 83% of drivers preferred the rear console location, according to their responses in the Driver Gesture Preference Questionnaire, represented as v. By combining these results, we calculated the scores of the two positions using Equation (2).

Ultimately, the rear console placement scored 90, while the center panel location scored 83. Therefore, we decided to position the camera in the rear console for the best user experience.

$$score = 100 \times (\frac{t_a - \bar{t_a}}{\sigma^2} \times v) \tag{2}$$

4.1.2. Selection of the Most Suitable Gesture

To select the most suitable gesture from the 30-gesture collection, we evaluated each gesture based on both objective and subjective indicators. As we have determined the camera positioning on the rear console in the previous step, we used the experimental data of the camera on the rear console to screen the gestures.

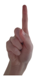
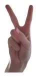
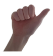
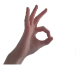
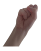
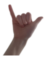
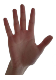
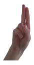
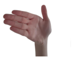
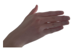
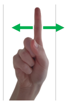
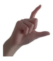
First, we calculated the average completion time of each gesture by normalizing the completion time data recorded in Experiment 1, represented as t. Then, we obtained the average score of each gesture from the driver's gesture preference questionnaire collected in Experiment 1, represented as s. Finally, we assigned equal weights to the objective and subjective indicators, and calculated the final score of each gesture using Equation (3). The results are presented in Table 7.

$$gs = 100 \times (0.5t + 0.5s) \tag{3}$$

4.1.3. Effects of Handedness

As previously stated, 4 out of 20 test subjects in the experiment are left-handed, making up 20% of the total sample. To investigate the effect of handedness on interaction efficiency, we analyzed the experimental data of the left-handed subjects separately from the data of the remaining 16 right-handed subjects. We used the Bonferroni test to compare the total time for gesture completion and the time for completing the gestures with the dominant hand for the two groups. The resulting probability value was p = 0.528, which exceeds the confidence level of 0.05. This suggests that handedness has no significant effect on interaction efficiency.

Table 7. Final gestures for the system.

Gesture	Function	Score
	Enter Music Control	98
	Enter Map Control	98
	Return	96
	Play Music	95
	Locate	94
	Answer the Call	94
	Pause Music	93
	Drag Map	93
	Previous Song	91
	Next Song	90
	Volume Change	85
	Zoom Map	85

4.2. Experiment 2

4.2.1. Primary Mission

In Experiment 2, we measured the primary mission using three indicators: driving penalty score, lane change path average deviation (MDEV), and lane change initiation (LCI). To obtain the driving penalty points for each driver, we summed and calculated the average of their penalties. The driving penalty points for the touch interaction method were consistently higher than those for the gesture interaction method in all tasks, as shown in Figure 10. The touch interaction method resulted in significantly higher driving penalty points than the gesture interaction method for music switching, music playback pause, and answering phone tasks, with only positioning and map zooming showing similar penalty points for both methods. We performed an ANOVA test and obtained the following results:

F(Switch Music) = 9.67, p = 0.001
F(Volume Adjustment) = 4.65, p = 0.01
F(Play &Pause Music) = 6.76, p = 0.004
F (Locate) = 3.77, p = 0.02
F(Zoom Map) = 2.768, p = 0.074
F(Answer Call) = 6.53, p = 0.006

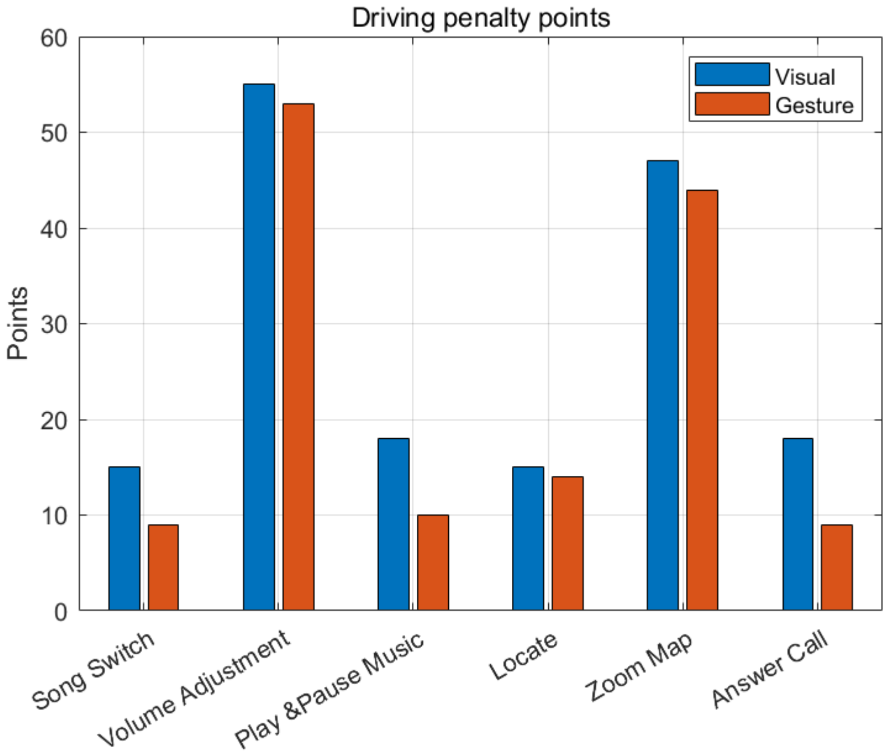


Figure 10. Driving penalty points. *V* stands for Visual Interface. *G* stands for Gesture Interface.

We used the Bonferroni test to establish a probability bound of 0.05, and found a significant difference in driving penalties between the touch and gesture interaction methods. The mean and standard deviation for all tasks are recorded in Table 8.

We also set up a control group of four drivers in the touch interaction mode, and the average driving penalty score for the control group was 0.75. This indicates that the difference in driving penalty score between the gesture interaction mode and normal driving conditions is small, indicating that the gesture interaction in the air does not significantly disturb the driver's driving behavior.

To analyze the MDEV and LCI indicators, we used the Bonferroni test. The corresponding data are shown in Tables 9 and 10. There were significant differences in the LCIs corresponding to different interaction modes in the music switching, music playback pause, and answering phone tasks. However, the differences in the MDEV metrics for these three tasks were not significant enough. This may be related to the selected experimental road conditions, which were mainly straight roads with fewer turning operations. Nevertheless, the LCI indicator reflects the driver's ability to respond to emergencies, which is critical for safe driving.

We also calculated the degree of improvement of the gesture interaction method compared with the touch interaction method. We used I to denote the degree of improvement, which can be calculated using Equation (4). Here, D_G and D_T represent driving penalty scores for in-air gestures and touch-based interactions, respectively. After calculations, we found that the average improvement when using in-air gesture interaction is 65%. This demonstrates that employing in-air gestures as an interaction method can significantly enhance driver performance in multi-goal scenarios, thereby increasing driver safety.

$$I = \frac{D_G - D_T}{D_T} \tag{4}$$

Table 8. Mean driving penalty points (*M*) and standard deviations (*S.D.*).

Interface	M_{SS}	$S.D._{SS}$	M_{VA}	$S.D._{VA}$	M_{PM}	$S.D._{PM}$
V	15	7.13	55	3.58	18	6.57
G	9	3.04	53	1.32	10	2.53
Interface	M_L	$S.D._L$	M_{ZM}	$S.D._{ZM}$	M_{AC}	$S.D._{AC}$
V	15	3.8	47	3.67	18	5.22
G	14	1.13	44	1.07	9	0.62

V stands for Visual Interface. *G* stands for Gesture Interface.

Table 9. Mean MDEV (*M*) and standard deviations (*S.D.*).

Interface	M_{SS}	$S.D._{SS}$	M_{VA}	$S.D._{VA}$	M_{PM}	$S.D._{PM}$
V	0.91	0.20	0.98	0.25	0.95	0.21
G	1.0	0.24	1.05	0.27	1.05	0.27
Interface	M_L	$S.D._L$	M_{ZM}	$S.D._{ZM}$	M_{AC}	$S.D._{AC}$
V	0.97	0.23	1.10	0.31	1.13	0.34
G	1.05	0.27	1.26	0.39	1.22	0.38

V stands for Visual Interface. *G* stands for Gesture Interface.

Table 10. Mean LCI (*M*) and standard deviations (*S.D.*).

Interface	M_{SS}	$S.D._{SS}$	M_{VA}	$S.D._{VA}$	M_{PM}	$S.D._{PM}$
V	−14.3	3.04	−14.1	3.45	−14.1	3.07
G	−13.5	3.84	−12.8	4.15	−13.1	4.38
Interface	M_L	$S.D._L$	M_{ZM}	$S.D._{ZM}$	M_{AC}	$S.D._{AC}$
V	−10.1	3.18	−10.5	4.41	−10.3	4.96
G	−13.0	4.23	−8.27	6.22	−8.21	5.07

V stands for Visual Interface. *G* stands for Gesture Interface.

4.2.2. Secondary Mission

In Experiment 2, we used the task completion time as a measure of the driver's completion of the control task on the IVIS system. The task completion time was counted from the time when the experimenter sent out an instruction to the time when the driver completed the corresponding control task in an interactive way. The application's prompt sound and pop-up window were used to judge whether the driver completed the task.

The average completion time of all tasks in the two interaction modes is shown in Figure 11. In most tasks, the completion time for the gesture interaction method was slightly longer than that for the touch interaction method. The time required for using the gesture interaction method was significantly longer than that for the touch interaction method in the tasks of zooming and volume adjustment.

We used the Bonferroni test to obtain the significance of all manipulation tasks as follows:

F(Song Switch) = 9.13, p = 0.002;
F(Volume Adjustment) = 7.95, p = 0.003;
F(Play &Pause Music) = 6.49, p = 0.004;
F (Locate) = 5.07, p = 0.03;
F(Zoom Map) = 4.39, p = 0.04;
F(Answer Call) = 9.86, p = 0.001.

Although there is a difference in task completion time, the Bonferroni test did not yield a significant difference between the task completion times of the two interaction modes.

The slightly longer task completion time observed in the in-air gesture interaction method compared to touch interaction may be attributed to the imperfect recognition accuracy of the in-air gesture interaction. Gesture recognition can also be affected by lighting conditions and obstacles. As a result, drivers may need to repeat the gesture multiple times to achieve the desired manipulation. Furthermore, the difference in task completion time is more noticeable in tasks that involve subtle manipulations, such as map positioning and volume adjustment, where the gesture interaction method may produce larger errors, leading the driver to make repeated attempts.

Despite the differences in task completion time, the performance of the two interaction modes in completing secondary missions is comparable. The observed differences are minimal and do not significantly impact task completion time.

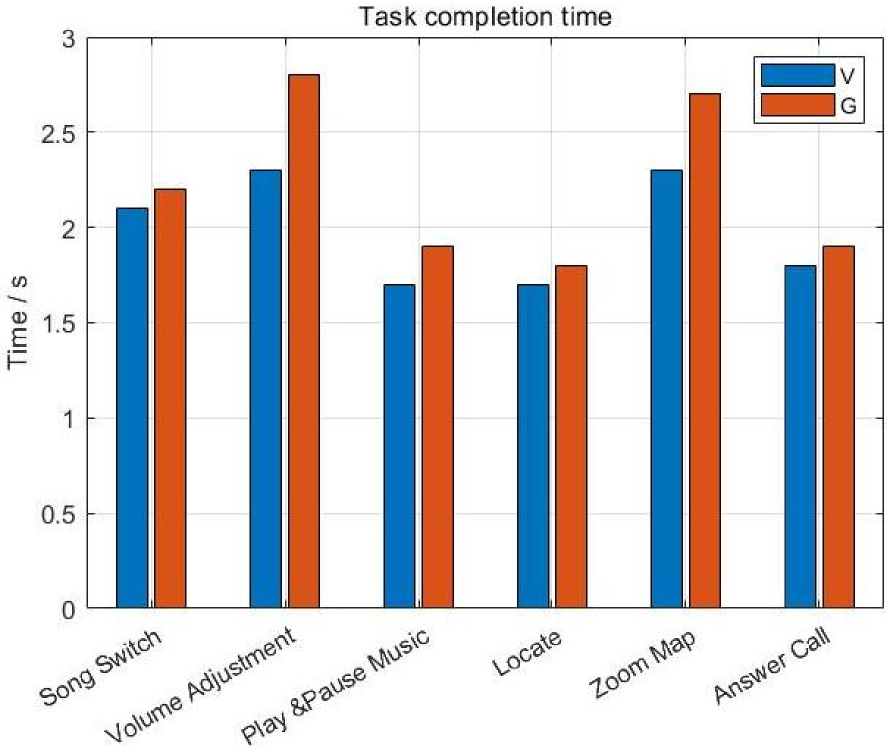


Figure 11. Task completion time. *V* stands for Visual Interface. *G* stands for Gesture Interface.

4.2.3. Subjective Evaluation

To evaluate the driver's perception of the interaction modes, we collected data on their workload and preference for each mode.

The results are presented in Figure 12, where we can see that the drivers rated the in-air gesture interaction method more favorably than the touch interaction method in terms of workload and preference.

To combine these two indicators, we normalized the data and calculated a percentile score for each interaction mode. The score for the touch interaction method was 75, while the score for the in-air gesture interaction method was 89. Therefore, according to the drivers' subjective evaluation, the in-air gesture interaction method is more preferred and better evaluated than the touch interaction method.

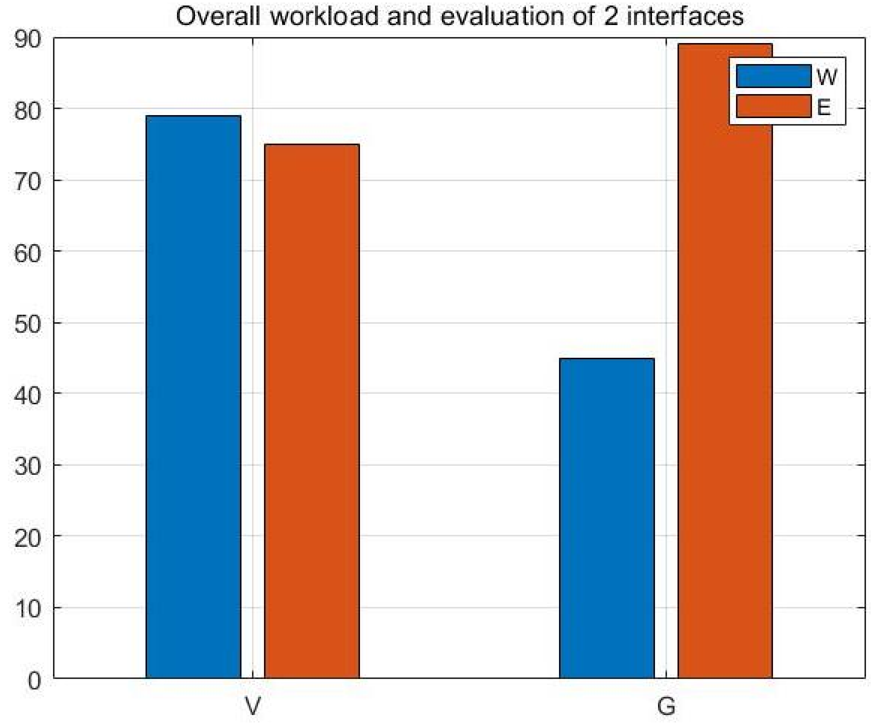


Figure 12. Subjective evaluations. *V* stands for Visual Interface. *G* stands for Gesture Interface. *W* means workload. *E* means evaluation.

5. Discussion

In the two conducted experiments, we developed and implemented the IVIS system, and evaluated the performance of the two interactive modes, touch and gesture.

5.1. Camera Positioning

In Experiment 1, we found that the time taken by the driver to make a gesture was shorter when the camera was placed on the rear console. We attribute this to the camera's recognition range and our skeleton-based gesture recognition scheme. The camera captures two-dimensional plane information, and when placed on the center panel, it recognizes parallel vertical planes. On the other hand, when placed on the rear console, it recognizes parallel horizontal planes, which aligns better with the driver's natural hand orientation in the car environment. Furthermore, our gesture recognition scheme calculates 2.5D coordinates of the palm key points, where 0.5D refers to the vertical distance from the camera. Since we use only an RGB camera and not additional sensors like in [45–47], we cannot detect the depth of the palm. Placing the camera on the center panel thus hinders the recognition of the palm's horizontal movement, which makes correct gesture recognition more challenging.

Therefore, we recommend placing the camera on the rear console for improved efficiency of gesture interaction, considering the limitations of the camera's recognition range and the skeleton-based gesture recognition framework.

5.2. Determination of the Most Suitable Gesture

In Experiment 1, we screened the gestures based on the combination of completion time and subjective evaluation. The scores of the last two gestures shown in Table 7 had a large gap with the scores of the previous gestures, because they were dynamic gestures that took longer to complete and were more susceptible to interference from light and obstacles in the car. Our experimental statistics showed a negative correlation between gesture completion time and subjective evaluation score, indicating that more complex gestures took longer to complete and received lower driver satisfaction ratings.

However, for tasks that require continuous control, dynamic gestures can improve the success rate of the operation. For example, the last two dynamic gestures in Table 7 were used for volume adjustment and map zooming, which require continuous change. If static gestures were used for these tasks, more operations would be required to complete the task.

Therefore, we believe that using static gestures for most functions in the IVIS system can ensure high interaction efficiency. For tasks that require continuous operation, using dynamic gestures can improve the accuracy of the operation and ensure high interaction efficiency.

5.3. Analysis of Interruption Caused by Gesture Interface

5.3.1. Primary Mission

In multi-target scenarios, the driver's primary mission is to steer the vehicle. We evaluated driving performance using three indicators: driving penalty points, MDEV, and LCI. In Experiment 2, the difference in MDEV between the two interaction modes was not significant, but for driving penalty points and LCI, there was a significant difference between the two interaction modes when completing tasks, such as music switching, music playback pause, and answering the phone. We believe that these tasks are relatively simple and require fewer cognitive resources. Therefore, the in-air gesture interaction allows drivers to quickly complete these tasks without looking at the screen and without interfering with their driving performance, resulting in significant differences between touch and in-air gesture interactions on the primary mission.

5.3.2. Secondary Mission

In multi-goal environments, the driver's secondary mission is mainly to complete in-vehicle tasks. We used task completion time for evaluation. In the analysis of the results of the secondary mission in Experiment 2, we found that although the gesture interaction method led to an increase in task completion time, this difference can be ignored. The system needs a certain amount of processing time for recognition of in-air gestures, but the processing time is acceptable and will not significantly increase the adjustment times or complexity of driver operations, and therefore does not demand too much of the driver's attention.

5.3.3. Subjective Evaluation

In Experiment 2, we integrated the driver's ratings of perceived workload and preference for the two gesture interaction methods to obtain their subjective evaluation. In terms of workload, the Mental Demands and Effort of the touch interaction method were significantly higher than those of the gesture interaction method. We believe that this is because when using the touch interface, drivers need to focus on the screen and determine the position of the operation button while paying attention to the road conditions. This can easily lead to increased psychological burden on the driver and make them more likely to become fatigued [35,44], which can harm driving safety.

6. Conclusions

In this paper, we introduce our proposed skeleton-based in-air gesture recognition framework and explore the application of in-air gesture interaction in the in-vehicle environment. Our framework divides in-air gestures into two categories and employs different methods to classify them. We conduct two experiments to study the performance of in-air gestures compared to touch interaction, with the goal of identifying the gestures that are most suitable for drivers and determining the optimal position for the camera.

Our gesture recognition model achieves an accuracy close to state-of-the-art and is optimized for our IVIS system, reducing resource utilization during the recognition process. This enhancement in gesture recognition efficiency for our IVIS system minimizes interference in subsequent experiments. We built a simulation environment and conducted two user experiments, establishing an evaluation system that indicates a 65% improvement in driver performance during in-car tasks when using gesture interaction compared to traditional interaction methods. Moreover, drivers experience lower workload stress when using gesture interactions.

In conclusion, our study confirms that in-air gesture interaction can improve the driver's driving performance and maintain the same interaction efficiency as touch interaction. We believe that the in-air gesture interaction method is well-suited for in-vehicle multi-goal environments, and has the potential to enhance the driver's experience and promote driving safety.

In the future, we plan to build upon the current effective gesture recognition framework in several ways. From a functional standpoint, we aim to increase the variety of recognizable gestures the system can detect and even support user-defined gesture recognition. From a technical perspective, we plan to incorporate additional interaction methods, such as lip reading and audio speech, as outlined in [8,15]. By using audio-speech recognition technology, we aim to further enhance the interactivity of the IVIS system we have designed.

Author Contributions: Conceptualization, C.C., G.S. and Y.W.; methodology, C.C., Y.X., H.D., W.Z. and X.K.; software, C.C. and W.Z.; validation, W.Z.; formal analysis, C.C. and Y.W.; investigation, C.C.; data curation, G.S.; writing—original draft preparation, C.C. and Y.W.; writing—review and editing, G.S. and X.K.; funding acquisition, G.S. and X.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003, in part by the National Natural Science Foundation of China under Grant 62073295 and Grant 62072409, and in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2022C01050.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The author declare no conflict of interest.

References

1. Bilius, L.B.; Vatavu, R.D. A synopsis of input modalities for in-vehicle infotainment and consumption of interactive media. In Proceedings of the ACM International Conference on Interactive Media Experiences, Barcelona, Spain, 17–19 June 2020; pp. 195–199.
2. Ba h, K.M.; Jæger, M.G.; Skov, M.B.; Thomassen, N.G. You can touch, but you can't look: Interacting with in-vehicle systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy, 5–10 April 2008; pp. 1139–1148.
3. Oviedo-Trespalacios, O.; Nandavar, S.; Haworth, N. How do perceptions of risk and other psychological factors influence the use of in-vehicle information systems (IVIS)? *Transp. Res. Part F Traffic Psychol. Behav.* **2019**, *67*, 113–122. [CrossRef]
4. Bulej, L.; Bureš, T.; Hnětynka, P.; Čamra, V.; Siegl, P.; Töpfer, M. IVIS: Highly customizable framework for visualization and processing of IoT data. In Proceedings of the 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Portoroz, Slovenia, 26–28 August 2020; pp. 585–588.
5. Kong, X.; Wu, Y.; Wang, H.; Xia, F. Edge Computing for Internet of Everything: A Survey. *IEEE Internet Things J.* **2022**, *9*, 23472–23485. [CrossRef]
6. Ryumin, D.; Kagirov, I.; Ivanko, D.; Axyonov, A.; Karpov, A. Automatic detection and recognition of 3d manual gestures for human-machine interaction. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W12*, 179–183. [CrossRef]
7. Jiang, S.; Sun, B.; Wang, L.; Bai, Y.; Li, K.; Fu, Y. Skeleton aware multi-modal sign language recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3413–3423.
8. Ryumin, D.; Ivanko, D.; Ryumina, E. Audio-Visual Speech and Gesture Recognition by Sensors of Mobile Devices. *Sensors* **2023**, *23*, 2284. [CrossRef]
9. Wu, Y.; Zheng, B.; Zhao, Y. Dynamic gesture recognition based on LSTM-CNN. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 2446–2450.
10. Kagirov, I.; Ryumin, D.; Axyonov, A. Method for multimodal recognition of one-handed sign language gestures through 3D convolution and LSTM neural networks. In Proceedings of the Speech and Computer: 21st International Conference, SPECOM 2019, Istanbul, Turkey, 20–25 August 2019; Springer: Berlin/Heidelberg, Germany, 2019, pp. 191–200.

11. Prabhakar, G.; Rajkhowa, P.; Harsha, D.; Biswas, P. A wearable virtual touch system for IVIS in cars. *J. Multimodal User Interfaces* **2022**, *16*, 87–106. [CrossRef]
12. Suh, Y.; Ferris, T.K. On-road evaluation of in-vehicle interface characteristics and their effects on performance of visual detection on the road and manual entry. *Hum. Factors* **2019**, *61*, 105–118. [CrossRef]
13. Kong, X.; Duan, G.; Hou, M.; Shen, G.; Wang, H.; Yan, X.; Collotta, M. Deep Reinforcement Learning-Based Energy-Efficient Edge Computing for Internet of Vehicles. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6308–6316. [CrossRef]
14. Ma, P.; Wang, Y.; Petridis, S.; Shen, J.; Pantic, M. Training strategies for improved lip-reading. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 8472–8476.
15. Ivanko, D.; Ryumin, D.; Kashevnik, A.; Axyonov, A.; Karnov, A. Visual Speech Recognition in a Driver Assistance System. In Proceedings of the 2022 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 29 August–2 September 2022; pp. 1131–1135.
16. Kim, M.; Yeo, J.H.; Ro, Y.M. Distinguishing homophenes using multi-head visual-audio memory for lip reading. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 22 February–1 March 2022; Volume 36; pp. 1174–1182.
17. Moon, G.; Yu, S.I.; Wen, H.; Shiratori, T.; Lee, K.M. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 548–564.
18. Sincan, O.M.; Keles, H.Y. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access* **2020**, *8*, 181340–181355. [CrossRef]
19. Li, D.; Rodriguez, C.; Yu, X.; Li, H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1459–1469.
20. Escalera, S.; Gonzàlez, J.; Baró, X.; Reyes, M.; Lopes, O.; Guyon, I.; Athitsos, V.; Escalante, H. Multi-modal gesture recognition challenge 2013: Dataset and results. In Proceedings of the 15th ACM on International Conference on Multimodal Interaction, Sydney, Australia, 9–13 December 2013; pp. 445–452.
21. Ronchetti, F.; Quiroga, F.; Estrebou, C.; Lanzarini, L.; Rosete, A. LSA64: A Dataset of Argentinian Sign Language. In Proceedings of the XX II Congreso Argentino de Ciencias de la Computación (CACIC), San Luis, Argentina, 3–5 October 2016.
22. Joze, H.R.V.; Koller, O. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv* **2018**, arXiv:1812.01053.
23. Tang, H.; Liu, H.; Xiao, W.; Sebe, N. Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. *Neurocomputing* **2019**, *331*, 424–433. [CrossRef]
24. Sagayam, K.M.; Hemanth, D.J.; Vasanth, X.A.; Henesy, L.E.; Ho, C.C. Optimization of a HMM-based hand gesture recognition system using a hybrid cuckoo search algorithm. In *Hybrid Metaheuristics for Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 87–114.
25. Yu, J.; Qin, M.; Zhou, S. Dynamic gesture recognition based on 2D convolutional neural network and feature fusion. *Sci. Rep.* **2022**, *12*, 4345. [CrossRef] [PubMed]
26. Lee, S.K.; Kim, J.H. Air-Text: Air-Writing and Recognition System. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, Chengdu, China, 20–24 October 2021; pp. 1267–1274.
27. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.L.; Grundmann, M. Mediapipe hands: On-device real-time hand tracking. *arXiv* **2020**, arXiv:2006.10214.
28. Dadashzadeh, A.; Targhi, A.T.; Tahmasbi, M.; Mirmehdi, M. HGR-Net: A fusion network for hand gesture segmentation and recognition. *IET Comput. Vis.* **2019**, *13*, 700–707. [CrossRef]
29. Guo, F.; He, Z.; Zhang, S.; Zhao, X.; Fang, J.; Tan, J. Normalized edge convolutional networks for skeleton-based hand gesture recognition. *Pattern Recognit.* **2021**, *118*, 108044. [CrossRef]
30. Novopoltsev, M.; Verkhovtsev, L.; Murtazin, R.; Milevich, D.; Zemtsova, I. Fine-tuning of sign language recognition models: A technical report. *arXiv* **2023**, arXiv:2302.07693.
31. De Coster, M.; Van Herreweghe, M.; Dambre, J. Isolated sign recognition from rgb video using pose flow and self-attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3441–3450.
32. Zha, Z.; Yuan, X.; Wen, B.; Zhou, J.; Zhang, J.; Zhu, C. From rank estimation to rank approximation: Rank residual constraint for image restoration. *IEEE Trans. Image Process.* **2019**, *29*, 3254–3269. [CrossRef]
33. Zha, Z.; Yuan, X.; Zhou, J.; Zhu, C.; Wen, B. Image restoration via simultaneous nonlocal self-similarity priors. *IEEE Trans. Image Process.* **2020**, *29*, 8561–8576. [CrossRef]
34. Zha, Z.; Yuan, X.; Wen, B.; Zhang, J.; Zhou, J.; Zhu, C. Image restoration using joint patch-group-based sparse representation. *IEEE Trans. Image Process.* **2020**, *29*, 7735–7750. [CrossRef]
35. Grahn, H.; Kujala, T. Impacts of touch screen size, user interface design, and subtask boundaries on in-car task's visual demand and driver distraction. *Int. J. Hum.-Comput. Stud.* **2020**, *142*, 102467. [CrossRef]
36. Vaezipour, A.; Rakotonirainy, A.; Haworth, N.; Delhomme, P. A simulator study of the effect of incentive on adoption and effectiveness of an in-vehicle human machine interface. *Transp. Res. Part F Traffic Psychol. Behav.* **2019**, *60*, 383–398. [CrossRef]

37. Jung, J.; Lee, S.; Hong, J.; Youn, E.; Lee, G. Voice+ tactile: Augmenting in-vehicle voice user interface with tactile touchpad interaction. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; pp. 1–12.
38. Kong, X.; Zhu, B.; Shen, G.; Workneh, T.C.; Ji, Z.; Chen, Y.; Liu, Z. Spatial-Temporal-Cost Combination Based Taxi Driving Fraud Detection for Collaborative Internet of Vehicles. *IEEE Trans. Ind. Inform.* **2022**, *18*, 3426–3436. . [CrossRef]
39. Gupta, S.; Bagga, S.; Sharma, D.K. Hand Gesture Recognition for Human Computer Interaction and Its Applications in Virtual Reality. In *Advanced Computational Intelligence Techniques for Virtual Reality in Healthcare*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 85–105.
40. Wong, A. NetScore: Towards universal metrics for large-scale performance analysis of deep neural networks for practical on-device edge usage. In Proceedings of the Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, 27–29 August 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 15–26.
41. Roider, F.; Raab, K. Implementation and evaluation of peripheral light feedback for mid-air gesture interaction in the car. In Proceedings of the 2018 14th International Conference on Intelligent Environments (IE), Rome, Italy, 25–28 June 2018; pp. 87–90.
42. Truschin, S.; Schermann, M.; Goswami, S.; Krcmar, H. Designing interfaces for multiple-goal environments: Experimental insights from in-vehicle speech interfaces. *ACM Trans. Comput.-Hum. Interact. (TOCHI)* **2014**, *21*, 1–24. [CrossRef]
43. Kong, X.; Chen, Q.; Hou, M.; Rahim, A.; Ma, K.; Xia, F. RMGen: A Tri-Layer Vehicular Trajectory Data Generation Model Exploring Urban Region Division and Mobility Pattern. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9225–9238. [CrossRef]
44. Sarter, N.B. Multiple-resource theory as a basis for multimodal interface design: Success stories, qualifications, and research needs. In *Attention: From Theory to Practice*; Oxford University Press: New York, NY, USA, 2007; pp. 187–195.
45. Li, H.; Wu, L.; Wang, H.; Han, C.; Quan, W.; Zhao, J. Hand gesture recognition enhancement based on spatial fuzzy matching in leap motion. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1885–1894. [CrossRef]
46. Liu, F.; Zeng, W.; Yuan, C.; Wang, Q.; Wang, Y. Kinect-based hand gesture recognition using trajectory information, hand motion dynamics and neural networks. *Artif. Intell. Rev.* **2019**, *52*, 563–583. [CrossRef]
47. Oudah, M.; Al-Naji, A.; Chahl, J. Elderly care based on hand gestures using Kinect sensor. *Computers* **2020**, *10*, 5. [CrossRef]

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