Multimedia Tools and Applications (2023) 82:12995–13023
https://doi.org/10.1007/s11042-022-13784-1

1224: NEW FRONTIERS IN MULTIMEDIA-BASED AND MULTIMODAL HCI

Multimodal user interaction with in-car equipment in real conditions based on touch and speech modes in the Persian language

Fateme Nazari [1] · Shima Tabibian [1] · Elaheh Homayounvala [2]

Received: 25 November 2021 / Revised: 13 June 2022 / Accepted: 5 September 2022 /
Published online: 19 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Nowadays, communication with in-car equipment is performed via a large number of buttons or a touch screen. This increases the need for driver's visual attention and leads to reduce the concentration of drivers while driving. Speech-based interaction has been introduced in recent years as a way to reduce driver distractions. This input mode faces several technical challenges such as the need to memorize voice commands and the difficulties of canceling them. This paper focuses on presenting a multimodal user interface design based on touch and speech modes, for controlling five in-car devices (radio, CD player or music player, fan, heater, and driver-side window). The research is designed to collect a dataset of in-car voice commands in the Persian language in real conditions (in a real car and in the presence of background noises) to firstly create a dataset of Persian voice commands (due to lack of research in this area in Persian speaking countries) and secondly intending to solve the mentioned challenges. To evaluate the proposed user interface, 15 participants performed ten different tasks based on the speech and touch modes, with and without driving simulation. The evaluation results indicated that the speech input mode with and without driving simulation has had in average smaller number of clicks for performing tasks (0.2 and 0.6), smaller task completion time (7.37 and 3.3 seconds), smaller time intervals between clicks (8.2 and 5 seconds) and smaller driver's distraction rate (25.08%) in comparison to the touch input

✉ Shima Tabibian
sh_tabibian@sbu.ac.ir

Fateme Nazari
fat.nazari@mail.sbu.ac.ir

Elaheh Homayounvala
e.homayounvala@londonmet.ac.uk

[1] Cyberspace Research Institute, Shahid Beheshti University, Shahid Shahriari Square, Daneshjou Boulevard, Shahid Chamran Highway, Tehran 1983969411, Iran

[2] School of Computing and Digital Media, London Metropolitan University, London, UK

mode, respectively. Moreover, using two different input modes in designing the in-vehicle user interface leads to increased accessibility.

Keywords In-vehicle equipment · Multimodal user interface · Voice command detection · Hidden Markov model · Accessibility

1 Introduction

Nowadays, multimodal interactions have been done based on different input modes such as speech, touch, gesture and, etc. [1, 9, 36]. The clear, flexible, accessible, and efficient interaction has made designers increasingly interested in multimodal designs [16]. Today, multimodal interactions are used in a variety of contexts including computers, cell phones, and smart devices [18, 30]. One of the smart devices in which multimodal interactions can be used is in-vehicle equipment. Secure in-car equipment control is a big challenge [26]. This control can be done by pressing a button, touching a screen, or via voice commands. Although the use of mobile applications such as social media or internet access while driving is generally prohibited, drivers are willing to use them while driving [26]. Thus, controlling the cars becomes more complicated and more distracting [26]. Although the in-vehicle infotainment system (IVIS) is the source of driver distraction, they are becoming more common every day [32]. The most important thing causing the drivers' distraction while driving is competitive tasks. Competitive tasks mean everything that conflicts with safe driving and distracts drivers [34]. Examples include reading a map, adjusting the radio, finding a song on a playlist, or searching for a location in a GPS navigation program while driving. These tasks may require considerable attention and take a lot of time from the driver. Therefore, choosing an efficient communication method can lower the interaction time and as a result reducing the driver's distraction. Using speech instead of touch or pressing buttons, in addition to its popularity, causes drivers to be less distracted. As a result, it will play a significant role in reducing the number of accidents [26]. However, it has different challenges such as the need to memorize voice commands, the difficulties of canceling them, the lack of proper datasets to train the voice command detection system, especially in some languages and the simulation and not the real environment of the recorded commands of the existing datasets. Some of these challenges are addressed in this paper. However, the lack of proper datasets to train the in-vehicle voice command detection system in some languages is serious. One of these languages is Persian which the mother language of the authors' of current paper. This is one of the main motivations of authors for choosing the Persian language. Although, there are some Persian datasets and software such as CPHPD (a Cell Phone Based Persian Digit Dataset exploited specially for spoken digit recognition in Persian phone dialers) [24], PVC_HSA (A Persian dataset of speaker-independent voice commands for controlling the smart home appliances) [15] and Nevisa (a software solution for Persian voice dictation) [28], there are no powerful speech-based interfaces such as Alexa, Siri or Cortana in Persian language. Additionally, according to the different nature of the Persian language in comparison to the English language, it is not a suitable way to adapt the mentioned speech-based interfaces to be exploited for the Persian language. Thus, as a starting step in the field of in-vehicle voice command detection for the Persian language in automotive industries of Persian speaking countries, we decided to collect a dataset of in-car voice commands in the Persian language in real conditions, in a real car and, in the presence of background noises.

The rest of the paper is structured in this way. In the second section, a review of the related works in the field of multimodal interaction with in-car equipment has been presented. The proposed approach of this paper will be discussed in section three with three sub-sections: the proposed Persian dataset of voice commands, the proposed voice command detection system, and the proposed in-car multimodal user interface. The experimental results will be presented in the fourth section. Finally, the paper is concluded in the fifth section.

2 Related works

In recent years, speech has also been considered by researchers as an input mode to control the user interface [3, 22, 26]. Speech-based interfaces such as Alexa, Siri, Cortana, etc. are increasingly exploited in everyday human life. Such interfaces are expected to become more common in cars in the future [3]. Speech-based interaction is a secondary task that should not negatively affect the primary task of driving [26]. If we want to compare the user interaction in the mobile environment with the user interaction in the car environment [11], there are several additional limitations in the car according to the need for high-level attention to driving. Therefore, there is a need for a system that minimizes the driver's distraction and provides intelligent access to complex and diverse information [11]. In recent years, especially in the last decade, several research studies have been concentrated on investigating issues and various methods to reduce drivers' distractions when working with in-car equipment using touch and speech modes, some of which are mentioned in the current work.

The live stream display of the road was located at the top of the car's touch screen. The aim was to ensure that drivers did not lose sight of the road when working with the touch screen. Through this live stream display, the drivers can see both obstacles on the road and do their tasks with the touch screen. Thus, live stream display help reduce drivers' distractions when working with in-car equipment. The main purpose of Buchhop et al. [6] was to answer three questions. The first question is about the value of the live stream display of the road above the car's touch screen. The second one relates to the ability of live stream in helping the driver to more easily detect obstacles while driving. And the last one asks about the possibility that the live stream performance plays an effective role in reducing driver attention. To answer these questions, several experiments were performed. The results showed that the live stream broadcast method could not reduce the distractions [6]. Various studies have been performed to eliminate these distractions. P. Green has conducted a study [12] to obtain the drivers' viewing time through the user interface, which was no more than 1.2 to 1.5 seconds. Then, he measured the time took to enter the destination which was between 1 and 2.5 seconds. In addition, a separate experiment showed that the 15-second rule is a certain amount for each task [12]. Another study [19] reviewed the above experiment to improve these results with only one difference; the time that is needed for braking was categorized into three situations (expected, unexpected, and surprising). The required time to respond to unexpected events was less than one second. The time of looking at the car interface has also been computed. The results showed that if the mentioned time was more than 1.6 to 2 seconds, it is considered a threat to safety [19]. In a study at the university in Finland [20], an experiment was conducted to compare speech, touch, and handwriting input modes in the in-car user interface. The experimental results showed that the speech input mode had the least amount of distraction for the driver [20]. The percentage of distraction when performing tasks using the speech mode was significantly lower than that of the touch mode as well as the handwriting mode [20]. The

percentage of distraction when performing tasks was 3.51 and 13.22 using speech and touch modes, respectively [20]. Another study [7] tries to control the driver's speed by adapting the music in a semi-conscious way in such a way that the sound of the music does not negatively affect the driver's performance. Additionally, it ultimately helps to reduce the driver's distraction. The results showed that the above technique has a positive effect on safe driving. As a result, it leads to reduce drivers' distractions [7]. On the other hand, several research studies have been conducted to reduce the driver's attention span using the speech mode. Among them is a study that asked users to use speech in two ways; uttering the whole voice commands and expressing just the main keywords of each command [5]. The experimental results showed that uttering the voice commands in the form of their keywords reduced the driver's cognitive load and did not have a negative effect on the driving performance [5].

As it can be seen in the literature, speech is a suitable mode of interaction and reduces the drivers' distraction. However, it has different challenges as addressed in the introduction section. The lack of proper datasets to train the in-vehicle voice command detection system in Persian language and the lack of a powerful in-vehicle voice command detection systems in automotive industries of Persian speaking countries were the main motivations of authors whom Persian is their first language, to choose Persian language in this study. Thus, as a starting step in the field of voice command detection in smart vehicles, we decided to collect a dataset of in-car voice commands in the Persian language in real conditions, in a real car and, in the presence of background noises. Moreover, an intelligent in-car multimodal interface in the Persian language is designed and developed based on a combination of the touch and speech input modes, which allows the driver to select their preferred input mode. Some of the challenges in the field of speech, such as considering different words, terms and, synonyms when using the speech mode, the user's need to memorize the commands, the ability to cancel tasks, and interaction without considering environmental conditions are addressed in the current research.

3 The proposed approach

In this section, we discuss our proposed approach in three sub-section. First of all, the proposed Persian dataset is discussed. Then, we present the proposed voice command detection system. Finally, the proposed in-car multimodal user interface has been explained.

3.1 The proposed Persian dataset

In this section, we will discuss the design process and recording conditions of the Persian voice commands for controlling in-car equipment (PVCCE).

3.1.1 Recording conditions

Data recording for PVCCE is done under a mobile phone, using Voice Recorder software, which is one of the Android applications. A series of initial settings are conducted to coordinate the collected data, before speakers' voices are recorded. This includes the following: speakers' voices are recorded in wave format, mono and at a bit rate of 16 kHz.

The settings of most available speech datasets are usually mono and not studio which means it is a single band and not two bands. Spoken files can be recorded in wave, mp3, or

other audio formats. However, since the wave format is more common than other formats, we have recorded our spoken files in the wave format. Sample rates between 8 kHz and 44.1 kHz can be considered in speech processing research. However, since a sample rate equal to 16 kHz is more common in speech datasets, we have recorded the voice commands with bitrate equal to 16 kHz. Although a higher sample rate increases the quality of the recorded speech signals, it will lead to an increase the computational complexity which does not worth the small amount of accuracy that may be achieved using higher quality speech signals.

PVCCE includes voice commands for turning on and off the player, radio, heater, and fan, as well as increasing and decreasing the player and radio sound and increasing or decreasing the heater and fan degree, as well as opening and closing the driver's side window. The whole dataset consists of 72 commands, 14 keywords (player, radio, heater, fan, window, on, off, play, disconnect, low, high, open, close, cancel) and 6 non-keywords (driver, side, left, forward, be, to be). The speakers (participants) were selected from drivers with different levels of computer literacy, different ages, uniformly distributed gender, different accents and different educational levels. The speakers are 20 people with the uniform gender distribution and with an age range of 18 to 45 years old and different educational levels. Six people have an age range of 18 to 28 years old, 11 people have the age range of 29 to 39 years old and three people are in the age range of 40 to 45 years old. The complete information of the speakers is shown in Table 1.

Each wave file is named according to its command abbreviation and is stored in a folder named sp. (speaker abbreviation) plus the speaker ID, which is a number between 1 and 20. For example, the "Play the radio" command is named (Pra.wav) and is stored in the folder related to each speaker. In Table 2, the English translation of the recorded commands has been presented. These commands are expressed individually by each speaker. For example, "Turn on the radio", "Make the radio to be turned on" and "Radio! On" are the translated forms of "رادیو روشن شود", "رادیو روشن شو" and "رادیو روشن", respectively.

Table 1 Data of Speakers (PVCCE)

Speaker ID	Gender	Age	Educational level	Persian accent
1	Male	40	Bachelor of science	Lori
2	Male	29	Bachelor of science	Araki
3	Male	35	Bachelor of science	Gilani
4	Male	29	Master of science	Lori
5	Male	32	Bachelor of science	Torki
6	Male	34	Bachelor of science	Shirazi
7	Male	36	Master of science	Neyshabouri
8	Male	28	Diploma of science	Kermani
9	Male	26	Bachelor of science	Lari
10	Male	41	Diploma of science	Baboli
11	Female	26	Bachelor of science	Tehrani
12	Female	45	Bachelor of science	Karaji
13	Female	18	Diploma of science	Tehrani
14	Female	32	Master of science	Gorgani
15	Female	30	Master of science	Tehrani
16	Female	33	Bachelor of science	Mashhadi
17	Female	30	Bachelor of science	Lori
18	Female	38	Bachelor of science	Torki
19	Female	27	Bachelor of science	Lori
20	Female	27	Diploma	Tehrani

Table 2 Introduction of recorded commands

Device	English translations of Persian Commands used in the study
Radio/Player	Turn on the radio/Make the radio to be turned on/ Radio! On
	Play the radio/Make the radio to be played / Radio! Play
	Turn off the radio/Make the radio to be turned off/ Radio! Off
	Disconnect the radio/Make the radio to be Disconnected/Radio! Disconnect
	Increase the radio volume/Make the radio volume to be increased/Radio! Increase the volume
	Decrease the radio volume /Make the radio volume to be decreased/Radio! Decrease the volume
Fan/Heater	Turn on the fan/Make the fan to be turned on/ Fan! On
	Turn off the fan/Make the fan to be turned off/ Fan! Off
	Increase the fan degree/Make the fan degree to be increased / Fan! Increase the degree
	Decrease the fan degree /Make the fan degree to be decreased Fan! Decrease the degree
Window	Open the driver side window/Make the driver side window to be opened/ The driver side window! Open
	Open the left front window/Make the left front window to be opened/ The left front window! Open
	Close the driver side window/Make the driver side window to be closed /The driver side window! Close
	Close the left front window/Make the left front window to be closed /The left front window! Close

As shown in Table 2, in this study five in-car devices including radio, player, fan, heater, and driver side window have been considered. Since the radio and player commands are similar, we just mentioned the radio commands in Table 2. The same is true for the fan and the heater.

3.1.2 Editing and labeling

To edit the wave files, each recorded file was opened in the Cool Edit Pro software environment. Cool Edit Pro software is a completely professional software with advanced tools and unique features for recording audio from various inputs. In addition to the audio recording, this software has other features such as editing audio files, converting audio files, applying effects, supporting a variety of audio formats, displaying audio files, batch processing, etc. One of the features of Cool Edit Pro that sets it apart from other similar software is that it maintains the quality of audio files without compromising any aspect. In order to edit the wave files, if sounds such as mouse click, sneeze, door-closing, etc. were present in the silent sections of the wave file, and they would be removed using the software. However, other noises such as the environmental noise (when the driver's side window is open), the car engine noise (in all commands), the radio noise (when the radio is on), the player noise (when the player is on), the heater noise (when the heater is on) and the fan noise (when the fan is on) would remain at the same intensity. It should be noted that since the data is recorded in the real condition, the amount of noise in the files is significant. By real condition we mean, recording drivers' voices in a real car and in the presence of motor, radio, and other background noises.

We have calculated the number of different noises in the wave files in terms of signal-to-noise ratio (SNR). It should be noted that all the speakers have spoken the command under these noisy conditions. Background noise for different speakers might be different. For example, consider the situation that the player, radio, or any other device is on and the speaker orders to turn it off. What the player/radio broadcasts for one speaker differs from another speaker. However, since this difference is not considerable, it was not considered. Therefore, in this section, we use one speaker as an example and calculate the amount of the different types of background noise in the wave files in terms of SNR.

To calculate the noise amount, the noise power in the silent sections of the wave files is obtained. Also, to calculate the clean speech power, the noise power in the silence part is subtracted from the noisy speech power in the non-silence parts. Then, the SNR is calculated according to the following equation.

$$\text{SNR} = 10 \times \log\left(\text{clean speech power}/\text{noise power}\right) \tag{1}$$

As shown in Table 3, the SNR is calculated for different types of noise. On average, the recorded files have an SNR of 23.5 dB. Most of the noise is related to the radio and player sounds. An example of one of the noisy files is given in Fig. 1.

Labeling is done manually at the word level using the Cool edit software. For the silent parts, the "sil" tag is used. The other parts of the commands are labeled according to their containing keywords and non-keywords (filler) parts. For example, for the command "Turn on the fan/تهویه روشن شود", the corresponding label file contains the following information:

0 3945000 sil
3945000 8345625 fan/تهویه
8345625 8545000 sil
8545000 12489375 on/روشن
12489375 17088750 filler/شود
17088750 21985625 sil

The range of words occurrence is determined at the sample level and converted to a unit of 10 microseconds using Eq. (2).

$$Time(10\mu s) = Samples \times \frac{10^7}{Sample\ rate(Hz)} \tag{2}$$

where the sample rate in this work is 16,000 Hz. Thus, the dataset (PVCCE) contains 1840 wave files (20 speakers that each performs 92 commands) with an average length of 2.7 seconds for each file (about one hour and 22 minutes for the whole dataset) and 1840 word-level label files.

3.1.3 The prepared dataset in comparison with other existing datasets

As explained in the introduction section, there is not any available dataset in the Persian language for controlling in-car equipment. However, there are several datasets in non-Persian languages. They contain different commands for turning on and off the accelerometer, the seat heating system, searching for a desired point on the map, closing the passenger side window, air conditioning, GPS, etc. [8, 35]. It should be noted that these commands are all recorded in the

Table 3 Calculation of the background noise amount in the recorded files for a sample speaker

Type of noise	SNR (dB)
Motor sound	40.25
Environmental sound	23.30
Radio sound	13.85
Player sound	10.15
Heater sound	25.30
Fan sound	28.15
Average	23.5

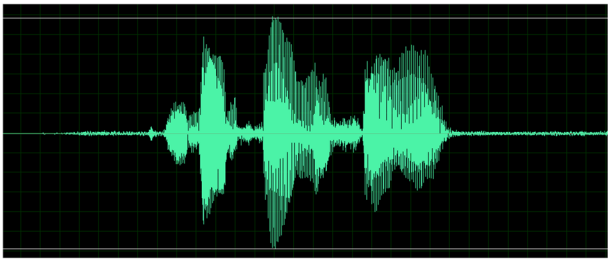


Fig. 1 Existence of noise in silence parts

simulation and not the real environment. While in this paper, all voice commands are recorded in real conditions (in all cases, the car is on). For example, when the driver order to close the driver's side window, the window was initially open and vice versa. Or when the driver order to turn off the radio, the radio is already on and vice versa. For all speakers, these conditions are fully established and they have recorded their commands in the real conditions. Another feature that is considered in the current work is solving some of the challenges of a voice command detection system. First, the driver does not need to memorize the commands, which means that if the driver wants to turn on the radio, he does not have to exactly use the "turn on the radio" command. Instead, if the driver just says the keywords "radio" and "on", the corresponding command will be detected. Second, the driver can turn on the radio using different commands, for example, to turn on the radio he/she can say "turn on the radio", "make the radio to be turned on" and "Radio! On". Third, if the driver uses his (her) command incorrectly, he (she) can cancel that command by saying just the "cancel" command and recall the correct command. These characteristics solve some of the most important challenges for in-car voice command detection systems.

3.2 The proposed voice command detection system

As shown in Fig. 2, the feature extraction and classification sections are the basic and important parts of a voice command detection system. We will introduce them, in the following paragraph.

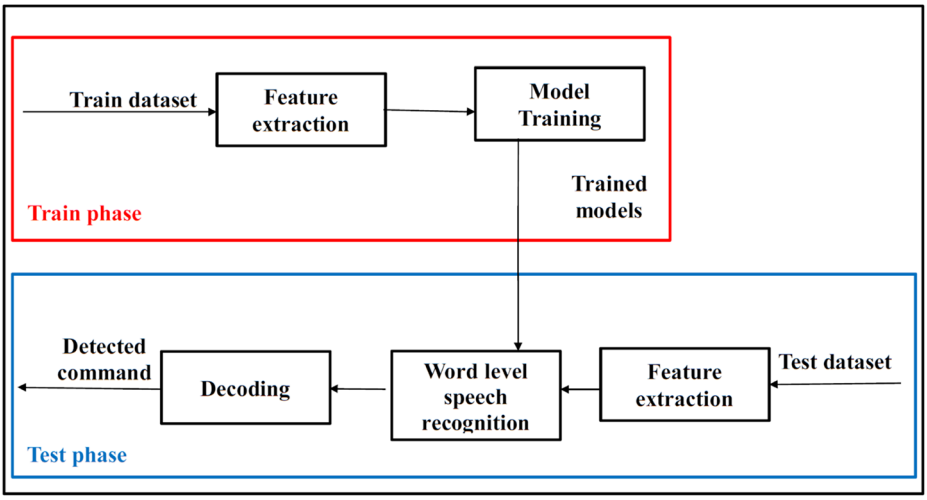


Fig. 2 The main parts of the voice command detection system

As shown in Fig. 2, the voice command detection system consists of two parts: train and test phases. In both the train and test phases, feature extraction is performed first. There are various methods for extracting features, including Linear Predictive Cepstral Coefficients (LPCC), Perceptual Linear Prediction (PLP), Mel Frequency Cepstral Coefficients (MFCC), and wavelet-based features [2, 14] . MFCCs are based on the human auditory and perceptual system and have also shown acceptable performance in various studies [14]. Therefore, MFCCs have been used as acoustic features, in this paper.

Then in the training phase, Hidden Markov Models (HMMs) are trained at the word level. The most common way to model speech from the past to the present is the HMM. Because of its properties, the HMM is a very good option for modeling the time-varying nature of speech signals. In recent years, neural networks have also been used to model speech signal [4]. Today, neural network-based methods have surpassed HMM-based methods in terms of accuracy [33]. However, these methods require a large amount of data for training. The field of voice command detection usually makes sense for a particular application; and in a particular application, the data volume is usually not very large. Therefore, we decided to use the HMM-based methods for training the word-level models.

In the test phase, after the test dataset is entered, feature extraction is performed. Then the word-level speech recognizer will produce the word-level recognition results using the trained models of the training phase. In the last step, the word-level results are entered into the decoding block and the decoded voice commands will be produced according to their compound keywords.

3.3 The proposed in-car multimodal user interface

Multimodal user interfaces are quite good in the provision of accessibility. One of the purposes of this paper is to provide a multimodal interface for the driver to let him/her select an input mode according to their conditions. Input modes for the proposed in-car user interface are touch and speech. One of the most important characteristics of user interfaces in the human-computer interaction (HCI) domain is usability. Usability means how well users can interact with the system to perform their tasks [29]. In this work, the user interface design was done using MATLAB AppDesigner environment.

We used the user-centered design methodology. In the user-centered design, users are involved in all of the stages of user interface designing such as user analysis, task analysis, application domain analysis, producing prototypes, evaluation, and producing the final product [10, 23] In designing the proposed multimodal user interface, firstly the users intended to use this interface (the target users) were identified. Secondly, the preconditions for achieving the desired goals of usability and user experience are determined. In this study, the target users are the drivers with different levels of computer literacy, different ages, uniformly distributed gender, different accents, and different educational levels. The identified users group was involved directly, mostly via several interview sessions, in all the design stages from the requirement identification phase to prototyping, implementation, and final evaluation. All the prototypes of our proposed user interface were checked and evaluated by the identified users in order to find possible problems and choose the best prototype. For example, the position of the user interface items, the color contrast, the distance of the element from each other, the visibility of the interface and other usability and user experience aspects were checked with the users. Among different user interface designs, the best prototype was selected for the next stage (implementation and final evaluation) based on comparative users' viewpoints. In order

evaluate the final proposed multimodal user interface, several tasks were determined with the necessary steps to perform them. The conditions of the final evaluation are discussed in section 4 of the paper. After going through various design techniques and different interviews with drivers, the final design of the car user interface was obtained as shown in Fig. 3.

As it is clear from Fig. 3, the proposed user interface has been designed according to the simplicity principle. Simplicity means taking away anything that is extra in the user interface [25]. All components should be critically examined and those that are not necessary should be removed. Fonts, button shapes, and sizes should be considered the same in different parts of the user interface. In fact, the main principle of consistency [21] has to be considered as well. As shown in Fig. 3, these points are completely observed in the design of the in-car user interface. Moreover, the text size in the proposed user interface is equal to 20 and 16 points, for bigger and smaller texts, respectively. Therefore, the readability of the text from the driver distance (which is about 65 to 70 cm) has been considered in the proposed user interfaces.

As shown in Fig. 3, black color is used for the background. Although dark colors may not be very conservative, they do not impair the readability of the texts. Therefore, it is acceptable in terms of usability and shows a different kind of interface. The elements are specified both as icons and as text. By clicking on any of the icons, the driver enters the page related to that device and can perform the desired operation. If he/she wants to use the speech mode, he/she can click on the microphone icon and order his/her command. Another point in the design is the grouping of the elements. For example, as it is clear in Fig. 3, the player and the radio are in the sound system group. The heater and the fan are in the heating and cooling system group, and the window is placed, separately, in the driver-side window group. This grouping helps drivers understand the user interface faster (learnability and memorability principles of HCI) and makes it easier for drivers to work with. As explained at the beginning, we preferred to design an interface that can be used by all drivers, especially for those who have poor computer literacy. For this purpose, the user help is considered as an audio file in the user interface. By logging in to the home screen, drivers can click on the help icon and learn how to work with the system, completely. Thus, the drivers can use the system without worry. Finally, after using the system, they can log out by clicking on the exit icon. Due to the principle of simplicity, differences should be eliminated as much as possible. But in some cases, the differences between the elements must be shown. For this purpose, the appropriate contrast should be used to show the differences, well. Contrast can be represented in terms of seven

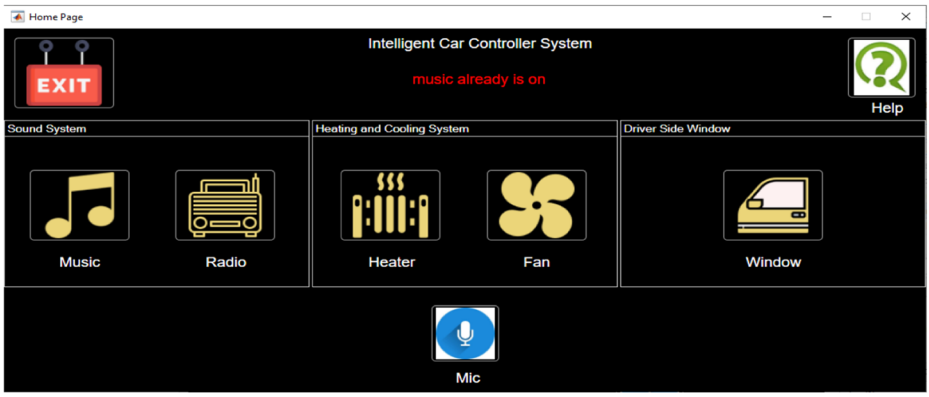


Fig. 3 The main page of the proposed in-vehicle multimodal user interface

visual variables of color, texture, shape, position, direction, size, value, or a combination of them [23].

In designing the in-car multimodal user interface, a combination of color, position, and value has been used to show the contrasts. For example, between "on" and "off" and "high" and "low" (Fig. 4). Because green is used to turn on and red to turn off the devices, illiterate drivers can do their job using color recognition. Position and value are important when the drivers are color-blind. As it is clear from Fig. 4, the position of the icon demonstrates "on" and "off". These two icons are in a group called "on and off", and as their group name indicates, the first icon is "on" and the second is "off". The value also means that the user can click on any of the icons to see the corresponding value (on or off) at the top of the page.

After designing the in-car multimodal user interface, it is necessary to establish the connection between the voice command detection system and the user interface. When the driver wants to communicate with the in-car multimodal user interface via speech mode, he can click on the microphone icon located on the main interface of the user interface. After that, the microphone is activated and records the input voice command. The input command will be sent to the voice command detection system and the decoded code of the input command will be returned to the user interface. Based on the sent code, the user's desired command is executed and is also displayed graphically in the user interface. For example, in the sound system group, when the driver commands the radio to be turned on, the radio will be turned on and the volume will be set to 50% by default. The output of the "radio on" command is displayed to the driver as shown in Fig. 5.

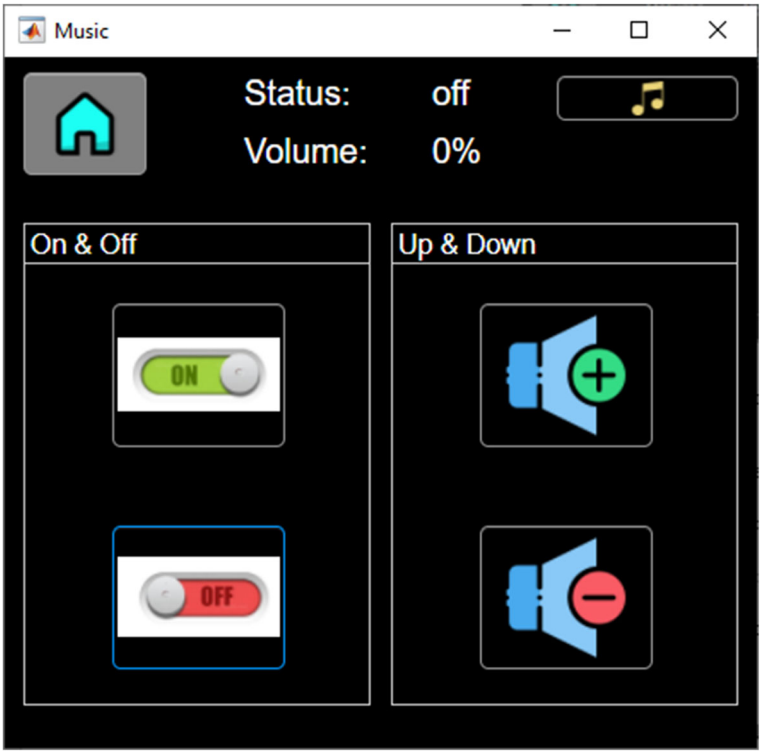


Fig. 4 Using visual variables to show contrast in the in-vehicle user interface

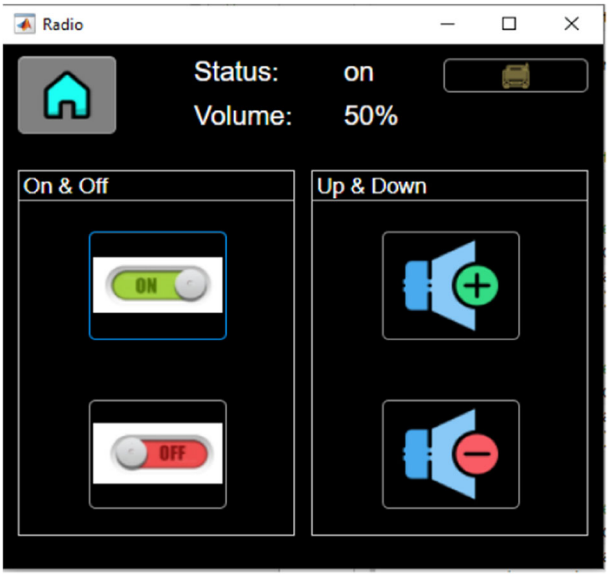


Fig. 5 Output of the smart user interface corresponds to the "radio on" command

If the driver wants to increase/decrease the volume of the radio, he/she order to increase/decrease the radio volume and the volume is set to 75/25%, respectively. If the driver wants to turn off the radio, he/she orders the radio to be turned off and the radio will be turned off. Tasks related to turning on/off, increase/decrease the volume/degree of the player, heater and fan are performed in the same way. For the driver's front window, the operations of opening and closing the window can also be done through the user interface. By default, the window is closed. Figure 6 shows the output of the driver's side window opening command (Closing is the same and only the status changes to "close").

In designing the in-car multimodal user interface, in order to anticipate the mistakes that the driver may make when using the speech mode to communicate with the user interface and to show the appropriate feedback to the driver. For example, the radio may be on and the driver may order the radio to be turned on. In this case, a message will be sent to the driver that "the radio is currently on". It may be claimed that when a device is on and playing, the driver hears it and does not make this error. Imagine a situation where the driver is listening to the radio while his/her cell phone is ringing. He/She mutes the radio to answer his/her call. When the driver finishes his/her talking on the cell phone, he/she may think the radio was turned off. Therefore, he/she orders the radio to be turned on. In this case, he/she should receive a message that the radio is on now. This error may even occur due to driver fatigue or other issues. This assumption is also considered for turning off. This message is defined for turning on/off all devices, including radios, players, heaters, and fans, and even opening and closing windows. For example, as you can see in Fig. 7, the error message is displayed in red in the center of the home screen to the driver. It should be noted that the system records the states of each device, independently. Thus, the new status of a device will not clear the previous status of other devices. As a result, if one device is on and the driver incorrectly orders the device to be turned on after executing some other commands, he/she will also receive the mentioned error message.

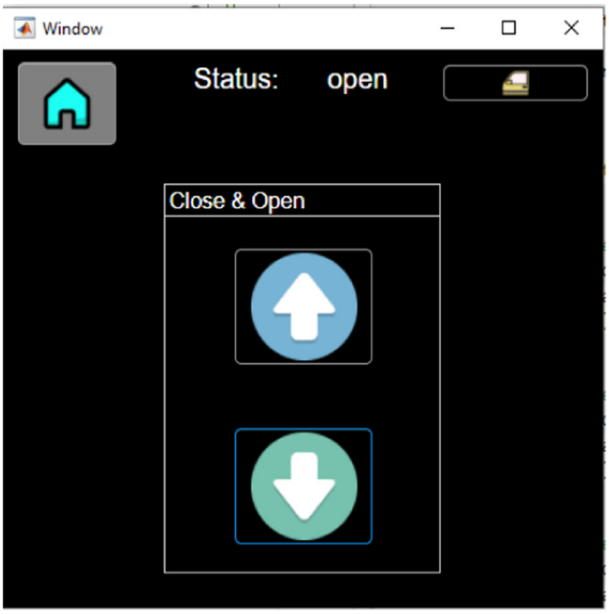


Fig. 6 Output of the in-car multimodal user interface corresponds to the driver's side window opening command

Since we considered two important aspects of usability, namely learnability and memorability, in designing the proposed in-car multimodal user interface, the time of looking at the car interface to perform different tasks is not more than 1.6 to 2 seconds. Therefore, according to [19], we can claim that the proposed in-car multimodal user interface is not a threat to driver's safety. As mentioned in the introduction section, the results of the study reported in [11] show that if the time for looking at the car interface is more than 1.6–2 seconds it is considered a threat to the driver's safety while doing both driving and working with the user interface. In this section, the main parts of the in-car multimodal user interface and its connection to the voice command detection system have been established. We will evaluate the proposed in-car multimodal user interface in the following section.

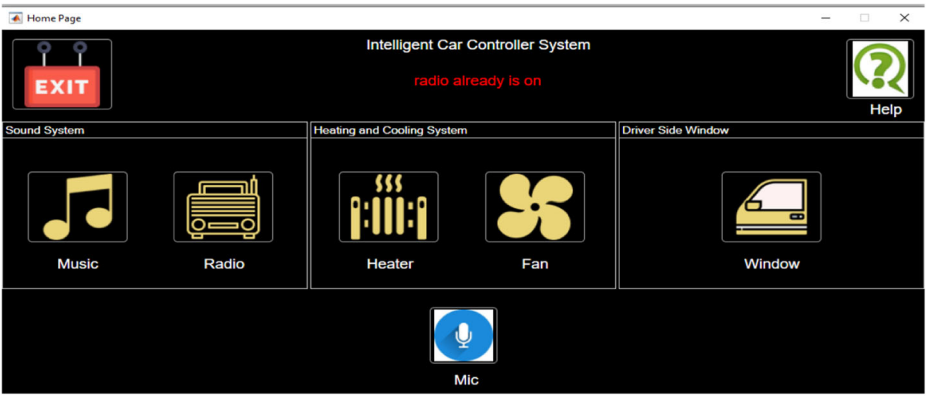


Fig. 7 Displaying an error message to the user when the commands are used, incorrectly

4 Experimental results

The word-level speech recognizer has been evaluated based on three criteria for evaluating the accuracy, correctness, and word error rate, which are obtained through the following equations [17]:

$$\text{Accuracy} = (\text{H}-\text{I})/\text{N} \tag{3}$$

$$\text{Correctness} = \text{H}/\text{N} \tag{4}$$

$$\text{WER} = (\text{D} + \text{S} + \text{I})/\text{N} \tag{5}$$

where H indicates the number of true detection. N indicates the total number of identifiable words. S, I and D indicate the substitution, insertion and deletion errors, respectively [31].

In order to evaluate the in-car multimodal user interface, four different experiments were performed to evaluate touch and speech modes with and without the driving simulation. The "with driving simulation" case is considered to model the cognitive load of driving. To conduct the assessments, 15 participants were invited to communicate with the user interface using both touch and speech modes. Then, an average was taken among all 15 participants. The evaluation criterion is calculated based on the number of clicks to perform the task, the time interval between clicks (in order to evaluate the speed of transition between different parts of doing a task), and the task completion time as well as the amount of driver distraction.

It should be noted that for evaluating the speech mode two other criteria (Real-time factor and true detection rate of the voice command detection system) have been exploited. The true detection rate of the voice command detection system is equal to the fraction of commands detected, correctly. It is calculated using an equation like Eq. (3). The Real-Time Factor (RTF) for commands that were correctly recognized at the first time, is calculated as follows.

$$\text{RTF} = \text{P}/\text{I} \tag{6}$$

where P is the processing time of the command with duration I. If the speaker's commands are not correctly detected for the first time but at the second time, the following equation is used for calculating the response time.

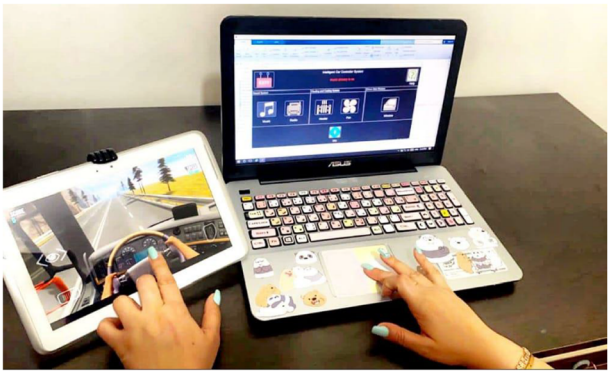
$$\text{RTF} = \text{X}_1/\text{Y}_1 \tag{7}$$

where X_1 is the cumulative processing time of the first and second time ($P_1 + P_2$). Y_1 is the cumulative time it takes for the speaker to express the command at the first and the second time ($I_1 + I_2$). The same is done for higher levels. Finally, if the RTF is less than one, the system has a real-time response speed.

In order to evaluate the levels of the distraction of the drivers, we ask the participants to play a driving game at the same time they communicate with the smart user interface. If they get an error in the game (for example a car accident) while working with the interface, it means that the interaction with the interface has caused their distraction. In this way, the error percentage is a good measure for the driver's distraction.

In this work, according to other literatures in this field [20, 22], the Intercity driving game is used to simulate the cognitive load of driving. Figure 8 shows pictures of the driving simulation environment. The driver's job is to drive while controlling the user interface using input modes (touch and speech). To make sure that the amount of cognitive load that the game

(a)



(b)

Fig. 8 Driving simulation environment: **a** a screen of Intercity driving game, **b** the simulation environment and the participant besides the car driving interface

imposes on the drivers is not much more than real driving, we installed the game on a tablet so that the driver can move it easily and feel more comfortable when working with it.

Among the all probable tasks (including turning on and off the player, radio, heater, and fan, as well as increasing and decreasing the volume of the player and the radio, and increasing and decreasing the degree of the heater and the fan and opening and closing the driver's side window), the process of turning on and off as well as increasing and decreasing the volume of the player and the radio is done in the same way. Additionally, both of these devices (radio and player) are in the same system group. Thus, we decided to use one of these devices (player) in the experiment. The same is true for heaters and fans. Before the participants wanted to complete the assigned tasks based on touch and speech modes, they had been provided with a sheet containing the following text.

"In this experiment, we intend to perform a series of tasks in the car using a multimodal user interface (an interface that communicates between a human and a device). Dear participant, you are asked to use the touch and speech mode in the car interface to perform tasks related to turning on and off the recording and heater, increasing and decreasing the recording sound and heater degree, as well as opening and closing the driver's side front window. Voice commands corresponds to the above tasks are shown in Table 4."

To measure the accuracy of the voice command detection system for all commands, 15 other speakers were invited to perform all the commands in real conditions of the car. For two of these 15 speakers, the accuracy of recognition was measured up to four repetitions of

Table 4 The complete tasks of four different experiments

Code	Task/Command
1	Turn on the player
2	Increase the player volume
3	Decrease the player volume
4	Turn off the player
5	Turn on the heater
6	Increase the heater degree
7	Decrease the heater degree
8	Turn off the heater
9	Open the driver side window
10	Close the driver side window

commands. For the other 11 speakers, the accuracy was obtained only for the first time. It should be noted that the time duration of the participant's familiarity with the user interface is a part of the experiment. Only a brief description of the user interface is given to each participant at the beginning and no information about the purpose of the experiment is provided to them.

Each participant communicates with the proposed in-car user interface through two modes of touch and speech with and without driving simulation. In the scenario without the driving simulation, the participant is only involved in the user interface and performs operations using touch or speech mode. In this case, the task completion time and the number of clicks to perform the task, as well as the time interval between clicks, can be obtained. Figure 9 shows pictures of this scenario.

As it is clear from Fig. 9, although we used laptop in order to evaluate the proposed multimodal in-vehicle user interface in real car and in presence of background noises (Fig. 9a), all the experiment could be done using tablet instead of laptop (Fig. 9b).

However, in the second scenario (with driving simulation), the participant interacts with the user interface, in addition to playing the driving game at the same time, we call this scenario "the scenario with the driving simulation". If he/she encounters an error in the game (accident) while working with the interface, it can be interpreted that he/she has become distracted. In this way, in addition to calculating the number of clicks to perform the task and associated task completion and the time interval between clicks, the driver distraction can also be measured. The next subsection describes the evaluation results of four mentioned experiments each in one sub-section.

4.1 Evaluation of the proposed dataset based on HMM

In the current work, the HMM is used to evaluate the proposed dataset (PVCCE). The dataset is divided into two parts: train and test. Two-thirds of the dataset has been selected for training and one-third for testing. The train and test sets are completely independent of each other based on their speakers. 1288 files of 14 speakers were used for training and 552 files of the other six speakers were used for testing.

The HMM Toolkit (HTK) [13] has been used to train word-level hidden Markov models. The number of hidden Markov models is 15 for 13 keywords, silence, and non-keyword parts of commands. The number of states of each model is considered, differently, from 8 to 16. In order to extract the features, each speech signal was divided into 25-millisecond frames with a 50% overlap using the Hamming windowing method. MFCC features have been extracted from each frame of the speech signal. We have used 12 MFCCs and one energy factor along with their first-order, second-order, and third-order derivatives to obtain a total of 39 or 52

(a)

(b)

Fig. 9 The first scenario in real car and in the presence of background noises: **a** our user study using laptop, **b** the possibility of using tablet instead of laptop

MFCC properties (based on computing the third-order derivatives or not). The number of Gaussian mixture functions at each state is assumed to be constant (16) for the different state numbers. The number of optimal states on these data has been determined after various evaluations. As shown in Table 5, the highest accuracy is obtained when the number of states is 12 and the lowest accuracy is obtained when the number of states is 18.

In another experiment, for each model, we consider 12 optimal states. However, in this case, the number of Gaussian mixture functions is chosen from 4 to 64). The evaluation results are presented in Table 6.

Table 5 Word-level HMM-based speech recognition evaluation results based on the number of different states of the hidden Markov model

Number of Gaussian mixtures	State number	Feature vector	Accuracy (%)	Correctness (%)	Word error rate (%)
16	8	39	81.43	83.87	18.57
16	10	39	81.88	83.27	18.12
16	12	39	82.36	83.60	17.64
16	14	39	82.12	82.97	17.88
16	16	39	79.82	80.73	20.18
16	18	39	78.68	79.07	21.32

Table 6 Word-level HMM-based speech recognition evaluation results based on the different number of Gaussian mixture functions in each state

Number of Gaussian mixtures	State number	Feature vector	Accuracy (%)	Correctness (%)	Word error rate (%)
4	12	39	80.55	81.66	19.45
8	12	39	80.58	81.58	19.42
16	12	39	82.36	83.60	17.64
32	12	39	81.97	83.45	18.03
64	12	39	79.98	81.73	20.02

In another experiment, for each model, we considered 12 optimal states and 16 optimal Gaussian mixture functions as constants and increased the feature vector size from 39 to 52. The evaluation results are presented in Table 7.

According to the results obtained from Tables 5, 6, and 7, it can be concluded that the best configuration for word-level HMM-based speech recognition on the PVCCE test dataset is the number of states of 12, the number of Gaussian mixture functions of 16 in each state and the number of MFCCs of 39. With this configuration the word–level HMM-based speech recognition has a correctness of 83.60%, an accuracy of 82.36%, and a word error rate of 17.64% based on the PVCCE test dataset.

4.2 Comparison of touch and speech modes without driving simulation

In this section, a comparison can be made between touch and speech without driving simulation based on the average evaluation results of 15 participants. The results of this comparison can be seen in Tables 8 and 9.

As shown in Tables 8 and 9, by evaluating the touch and the speech modes without driving simulation, we found that in the speech mode, the (number of clicks to perform the task is on average one click better than that in the touch mode. It is noteworthy that in the touch mode

Table 7 Word-level HMM-based speech recognition evaluation based on the number of MFCCs

Number of Gaussian mixtures	State number	Feature vector	Accuracy (%)	Correctness (%)	Word error rate (%)
16	12	39	82.36	83.60	17.64
16	12	52	81.97	83.96	18.03

Table 8 Performance measures for different tasks when using touch mode without driving simulation

Task code	Number of clicks	Time interval between clicks (S)	Task completion time (S)
1	3	11.74	13.25
2	4	13.15	15.41
3	3	12.2	14.86
4	2	7.03	8.12
5	2	7	8.05
6	3	10.26	12.58
7	3	10.22	11.60
8	2	6.08	7.33
9	2	7.30	8.84
10	2	7.04	8.10

without driving simulation, the user achieves his/her goal on the first try. However, for the speech mode without driving simulation, it is possible that the voice command detection system could not detect the user command on his/her first try. Sometimes (for a few cases) it happens that the command is not detected at all. Moreover, in the touch mode without driving simulation, the time interval between clicks to perform the task is about 5 seconds greater than that in the speech mode. In addition, in the touch mode without driving simulation, the user needs an average of 10.8 seconds to perform the task which is about 3 seconds greater than that in the speech mode. Another point obtained from Table 9 is the poor performance of the voice command detection system in detecting tasks 3 and 7. This is because of specific commands. These two tasks are "Decrease the player volume" and "Decrease the heater degree". One of the main keywords in these commands is decrease which is translated in Persian to " " which is pronounced as "kam". As it is clear, this keyword has a very short duration. Thus, it is very likely that it will be mistakenly replaced with another short-length keyword or filler in the recognition results. This leads to not detecting the corresponding command in the first step or even in the next steps. Figure 10 shows a comparison of the touch and speech modes without driving simulation in terms of the number of clicks to perform the task for 10 tasks.

As it is clear from Fig. 10, it is shown that the number of clicks to perform the task in the touch mode is greater than that in the speech mode for most of the tasks. Figure 11 shows a comparison of the touch and speech modes without driving simulation in terms of time interval between clicks to perform the task for each of the 10 tasks.

Table 9 Performance measures for different tasks when using speech mode without driving simulation

Task code	1st step detection (%)	2nd step detection (%)	3rd step detection (%)	Not detection (%)	Number of clicks	Time interval between clicks (S)	Task completion time (S)	RTF
1	93.33	–	6.67	–	3	9.11	11.38	0.61
2	93.33	6.67	–	–	2	7	8.1	0.60
3	33.34	13.33	40	13.33	4	10.55	12.5	0.62
4	100	–	–	–	1	0	4	0.57
5	93.33	6.67	–	–	2	7.03	8.12	0.59
6	100	–	–	–	1	0	4.23	0.61
7	46.68	26.66	13.33	13.33	4	10.2	12.2	0.67
8	100	–	–	–	1	0	4.15	0.61
9	100	–	–	–	1	0	5.5	0.47
10	100	–	–	–	1	0	5.3	0.47

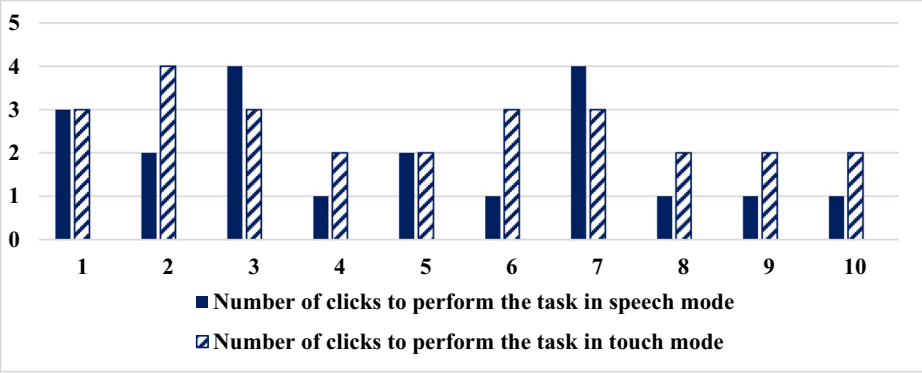


Fig. 10 Comparison of the touch and speech modes without driving simulation in terms of number of clicks

As it is clear from Fig. 11, the time interval between clicks to perform the tasks in the touch mode is greater than that in the speech mode for all tasks. Figure 12 shows a comparison of the touch and speech modes without driving simulation in terms of the task completion time for the 10 tasks.

As it is clear from Fig. 12, the average time required to perform the defined tasks in the touch mode is greater than that in the speech mode for almost all 10 tasks.

4.3 Comparison of the touch and speech modes with driving simulation

In this section, a comparison can be made between touch and speech with driving simulation based on the average evaluation results of 15 participants. The results of this comparison can be seen in Tables 10 and 11.

As shown in Tables 10 and 11, by evaluating the touch and the speech modes with driving simulation, we find that in the touch mode with driving simulation, the number of clicks to perform the task is in average one click more than that in the speech mode. It is noteworthy that in the touch mode without driving simulation, the user achieves his/her goal on the first try. However, for the speech mode, it is possible that the voice command detection system could not detect the user command on his/her first try. Sometimes (for a few cases) it happens

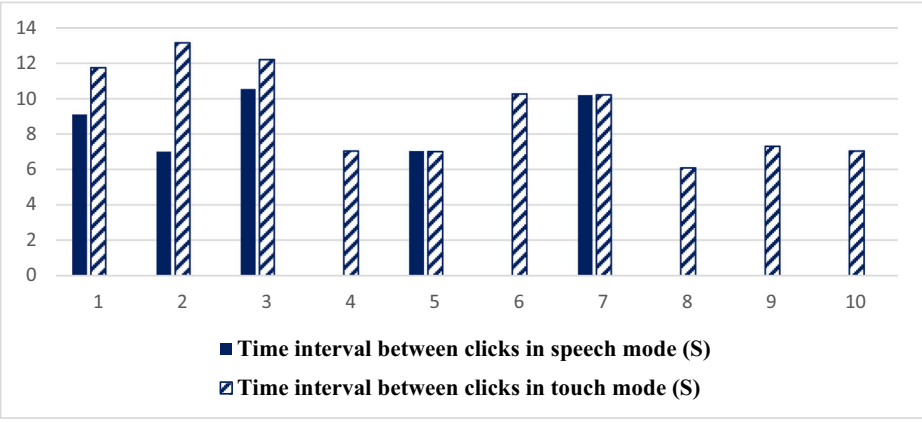


Fig. 11 Comparison of the touch and speech modes without driving simulation in terms of the time interval between clicks

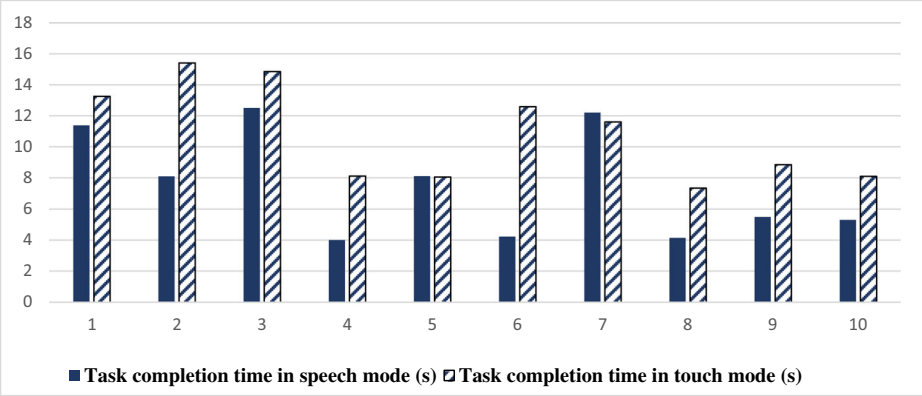


Fig. 12 Comparison of the touch and speech modes without driving simulation in terms of the task completion time

that the command is not detected at all. Moreover, in the touch mode with driving simulation, the time interval between clicks to perform the task is on average 8 seconds greater than that in the speech mode. Additionally, in the touch mode with driving simulation, the user needs an average of 16.81 seconds to perform the task. However, in the speech mode, the driver needs an average of 9.44 seconds to execute the command. By comparing the touch and speech modes with driving simulation, we find that the results are similar to the evaluation results of comparing the touch and speech modes without driving simulation. However, the purpose of having a driving game at this stage is to be able to measure the driver's distraction when using the touch and speech modes. As can be seen in Tables 10 and 11, the distraction rate in the touch mode with driving simulation is much higher than that of the speech mode. Again, as the results in Table 9, the poor performance of the voice command detection system in detecting tasks 3 and 7 is prominent according to the previous discussion. Figure 13 shows a comparison of the touch and speech modes with driving simulation in terms of the number of clicks to perform the task for the ten tasks.

As it is clear from Fig. 13, the number of clicks to perform the task in the touch mode is on average greater than that in the speech mode for most of the tasks. Figure 14 shows a comparison of the touch and speech modes with driving simulation in terms of the time interval between clicks to perform the task for the ten tasks.

Table 10 Performance measures for different tasks when using the touch mode with driving simulation

Task code	Number of clicks	Time interval between clicks (S)	Task completion time	Distraction rate (%)
1	2	15.08	19.47	33
2	3	22.12	24	60
3	4	20	22.41	20
4	2	12.42	14.72	26
5	2	13.5	15.90	13
6	3	15.15	18.22	53
7	3	10.05	12.66	13
8	2	13.1	15.20	13
9	2	11.14	13.60	20
10	2	10.23	12	13

Table 11 Performance measures for different tasks when using speech mode with driving simulation

Task code	1st step detection (%)	2nd step detection (%)	3rd step detection (%)	Not detection (%)	Number of clicks	Time interval between clicks (S)	Task completion time (S)	RTF	Distraction rate (%)
1	93.33	–	–	6.67	4	14.18	16.25	0.73	6.66
2	93.33	6.67	–	–	2	7.38	9.1	0.70	0
3	20	20	26.66	33.34	4	15.1	16.45	0.73	0
4	100	–	–	–	1	0	5.32	0.63	0
5	86.66	6.67	–	6.67	4	12.15	14.68	0.66	0
6	100	–	–	–	1	0	4.5	0.72	0
7	26.66	20	20	33.34	4	12.1	14.21	0.74	0
8	100	–	–	–	1	0	4.75	0.67	0
9	100	–	–	–	1	0	4.36	0.50	6.66
10	100	–	–	–	1	0	4.82	0.50	0

As it is clear from Fig. 14, the time interval between clicks to perform the tasks defined in the touch mode is longer than that in speech mode for almost all ten tasks. Figure 15 shows a comparison of the touch and speech modes with driving simulation in terms of the task completion time for the ten tasks.

As it is clear from Fig. 15, the average time required to perform the defined tasks in the touch mode is longer than that in the speech mode for all ten tasks. Figure 16 shows a comparison of the touch and speech modes with driving simulation in terms of the distraction rate to perform the ten tasks.

As it is clear from Fig. 16, the average distraction rate of performing tasks in the touch mode is significantly greater than that in the speech mode for all ten tasks.

4.4 Evaluations in real conditions

As mentioned at the beginning, by inviting 15 participants, the accuracy of the voice command detection system in real conditions is measured for all 72 commands. It should be noted that to maintain safety, the users are not driving during the experiments but the car engine is on. The

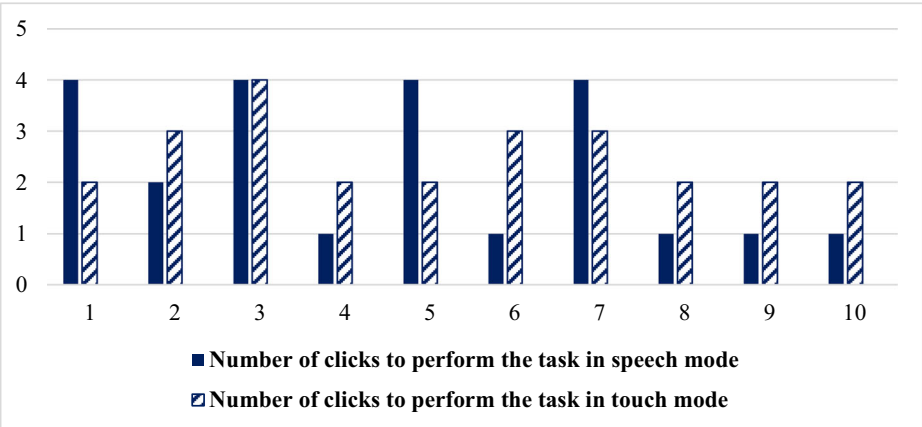


Fig. 13 Comparison of the touch and speech modes with driving simulation in terms of number of clicks

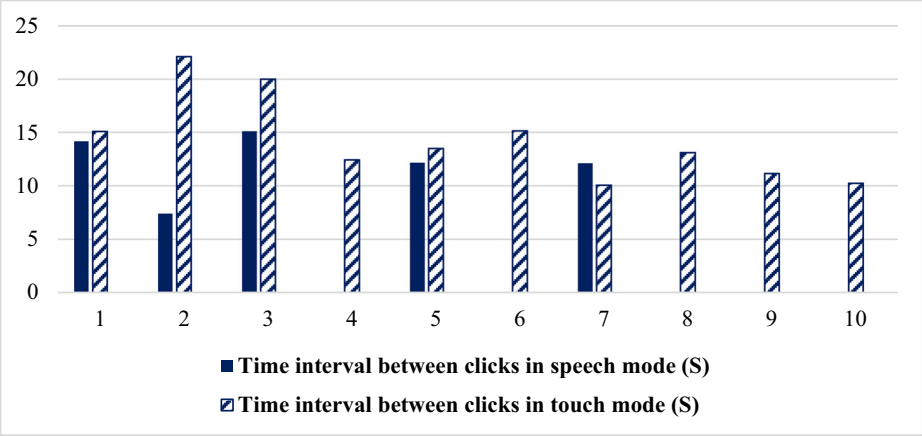


Fig. 14 Comparison of the touch and speech modes with driving simulation in terms of the time interval between clicks

distraction test has not been performed for these participants. For two participants, it takes four steps for the voice command detection system to fully detect all voice commands. The evaluation results have been presented in Table 12. For the other 11 participants, the detection of commands has been completed at the first step (Table 13).

As it is shown in Tables 12, 110 commands of 144 commands were correctly detected in the first step. In the second step, 16 commands of the 34 remaining commands were correctly detected. In the third step, 2 commands of the remaining 18 commands were correctly detected. In the fourth step, the results none of the 16 remaining commands were detected. Therefore, 16 commands of the whole 144 commands remain undetected after four attempts. As a result, the rate of commands that have not been correctly detected is 11.11%. After completion of all four steps, the voice command detection system has an accuracy of 88.89% in real conditions. Table 13 measures the accuracy of voice commands detection in just the first step for 15 speakers.

Out of a total of 1080 commands for 15 speakers, 886 commands were correctly detected at the first step, and 194 commands were not correctly detected. Thus, the true detection rate of just one attempt is about 82% in real conditions.

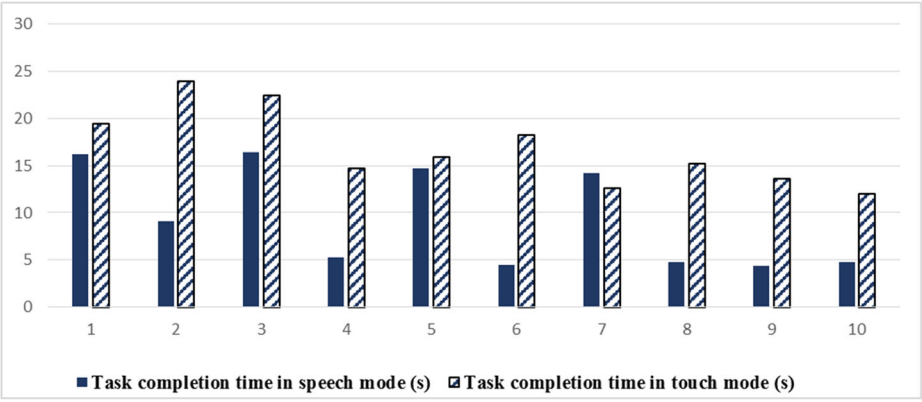


Fig. 15 Comparison of the touch and speech modes with driving simulation in terms of the task completion time

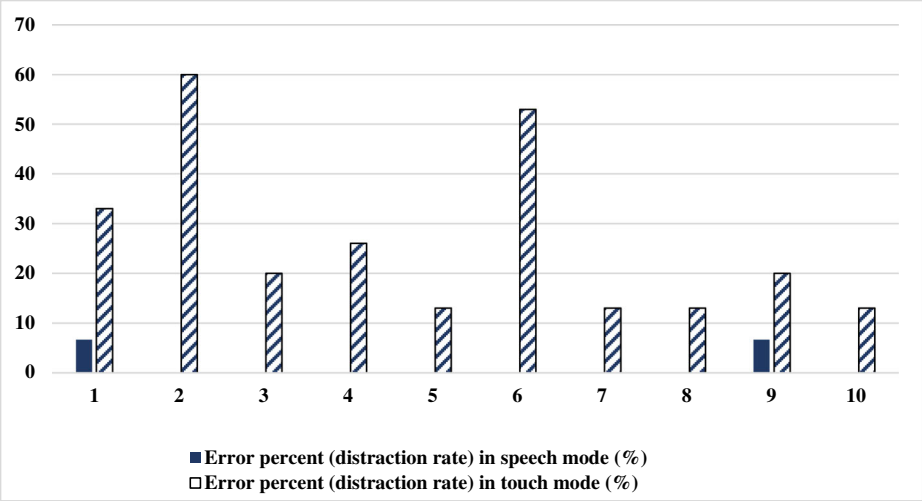


Fig. 16 Comparison of the touch and speech modes with driving simulation in terms of distraction rate

4.5 Comparison of the current work with the previous research studies

Table 14 compares the current work with previous research studies in the field of intelligent in-vehicle multimodal user interfaces.

As it is shown in Table 14, the current work is compared with the previous research studies. One of the differences between our work and previous works is that most of the researchers evaluated their results just based on distraction rate measures. Moreover, the number of considered tasks in the previous works is usually less than five. However, in our work, each participant performed 10 different tasks and the evaluation results were compared based on four criteria: distraction rate, task completion time, time intervals between clicks, and the number of clicks to perform the task. In the second research, which is shown in Table 14, the distraction rate of the participants by performing only three tasks in the speech mode was about 2% greater than that in our work. However, in the touch mode, the distraction rate in our work

Table 12 Accuracy of voice commands detection in four steps for two speakers

Step number	Detected commands (%)	Not detected commands (%)	RTF for detected commands	RTF for not detected commands
1	76.38	23.62	0.69	0.72
2	47.05	52.95	0.73	0.47
3	11.11	88.89	0.37	0.51
4	0	100	–	0.53

Table 13 Accuracy of voice command detection in the first step for 15 speakers

Number of speakers	Total number of commands	Percentage of detected commands (%)	Percentage of undetected commands (%)	Real Time Factor of commands)RTF)
15	1080	0. 82	0.18	0.49

Table 14 Comparison of the current study with the previous researches

Research	Paper year	Input modes	Number of tasks	Number of participants	Distraction rate (%)	Task completion time (S)	Number of clicks
The current paper	2021	Touch and speech	10	15	Touch 26.4, Speech 1.32	Touch 16.81, Speech 9.44	Touch 3 clicks, Speech 2 clicks
[20]	2017	Touch and speech	3	20	Touch 13.22, Speech 3.51	Not reported	Not reported
[6]	2017	Touch with live road broadcast and without live road broadcast	3	24	Touch with live road broadcast 6.94, Touch without live road broadcast 8.33	Touch with live road broadcast 23.33, Touch without live road broadcast 24	Not reported
[7]	2017	Touch with music fading and without music fading	Not reported	26	The distraction rate in the first case is less than that in the second case.	Not reported	Not reported
[27]	2019	Touch and speech	2	18	Not reported	Touch 5.40, Speech 5.35	Not reported

was 26.4%, which is higher (about 13%) than the reported distraction rate in the second research in Table 14 [20]. In the third research in Table 14 [6], the main purpose was to decrease the distraction rate by using a live road display above the touch screen; however, the performance of this technique has not been satisfactory. In the fourth research in Table 14 [7] an approach was proposed to improve the distraction rate in the touch mode by fading music. This technique has finally been approved. However, the evaluation results have not been reported based on any objective measure. In the fifth research reported in Table 14 [27], the time to perform tasks using the speech and touch modes was 5.35 and 5.40 seconds, respectively. In the current work, the time duration of performing the tasks using the speech and touch modes has been obtained equal to 9.44 and 16.81 seconds, respectively. Thus, the tasks completion time in our work is greater than that in the previous researches. This is according to this matter that in the previous works, a series of exercises were discussed for the participants to make them familiar with the system. While in our work, no information was given to users about the purpose of the experiment. Thus, in our work, the time of users' familiarity with the user interface has formed part of the time duration for finalizing the tasks. The time it takes for user to familiarize themselves with the user interface was on average about 5 seconds. If we subtract this value from the whole-time duration of performing the tasks in the speech and touch modes, we will find that our work is superior in terms of time duration in the speech mode compared to the previous works.

5 Conclusion

Multimodal user interfaces are quite suitable in the provision of accessibility. In this way, users with physical disabilities, such as hand trembling or lisping, can use the preferred input mode of the multimodal user interfaces according to their disabilities. One of the situation in which using multimodal user interfaces is very critical, is in the intelligent vehicles. Multimodal user interfaces in intelligent vehicles covers a wide range of drivers and provides good accessibility. Even the drivers who are physically healthy may use different input modes in different situations. For example, it is easier to work with a virtual map using the touch mode and the speech mode is more appropriate for navigation. Most of the studies in the field of multimodal user interfaces have considered the touch and speech modes as input modes. In order to consider speech as one of the input mode, it is necessary to have enough dataset for training the voice command detection system. Unfortunately, the lack of the Persian dataset in the field of in-vehicle voice control was an important challenge to train such voice command detection system. One of the aims of this paper is to provide an in-car multimodal user interface based on touch and speech input modes for Persian language so that the Persian drivers can select an input mode according to their conditions. Another challenge in the previous studies multimodal in-car user interfaces with speech mode is the lack of recorded data in the real environment of the car, the need for users to memorize commands, and the difficulty of canceling speech commands. Thus, we concentrated to solve the mentioned challenges in Persian Language. Another purpose of this paper was to compare the touch and speech modes based on the four criteria of number of clicks, task completion time, time intervals between clicks, and distraction rate for drivers using commands in the Persian language. Moreover, the designing of the proposed in-car multimodal user interface has been done in such a way that those two important aspects of usability (learnability and memorability) have been considered.

Thus, the time of looking at the car interface to perform different tasks is not more than 1.6 to 2 seconds which is lower than the threshold limit to be of any threat to drivers' safety.

Based on the evaluation results, it is shown that the number of clicks to perform the task defined in the touch mode is on average 3 clicks. This is one click more than the number of clicks in the speech mode. Additionally, it is shown that the amount of time required to perform the defined tasks in the touch mode is on average about 7 seconds longer than the time required to perform tasks in the speech mode. Additionally, the time interval between clicks to perform the defined tasks in the touch mode is on average about 8 seconds longer than that in the speech mode. Moreover, it has been shown that the distraction rate when performing the defined tasks in the touch mode is on average about 25% more than the distraction rate for performing tasks in the speech mode. This property is the most important strength of the speech mode and has been an active research field in the last decade.

In the end, it can be concluded that both speech and touch modes have their unique strengths and weaknesses. The availability of both modes can compensate for the disadvantages of a single and enable drivers to use the most appropriate interaction mode due to their situations.

Comparison of the current work with previous research studies indicates that firstly, most previous works evaluated their results only based on distraction rate criterion; while in this work, we evaluated the results based on four criteria: distraction rate, task completion time, the time interval between clicks and number of clicks. Moreover, for evaluating the speech mode two other criteria (real-time factor and true detection rate of the voice command detection system) have been exploited. Secondly, the number of tasks included in most of the previous works is usually less than five; while in our work, each of the participants performed 10 different tasks. Thirdly, in the current work, the distraction rate has been improved compared to the previous research studies. The most important difference is that we evaluated our work in both real and simulated conditions. However, the previous works have been evaluated just in simulated conditions.

In the current work, the voice command detection system has been trained based on the hidden Markov model. In future works, neural network-based methods can be used to increase the true detection rate of the voice command detection system. Additionally, we can use ontology to increase the accuracy of understanding concepts of the commands.

Data availability The datasets generated during the current study are not publicly available. However, further information about the data and conditions for access are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Aftab AR (2019) Multimodal driver interaction with gesture, gaze and speech. In: International conference on multimodal interaction, pp 487–492
2. Azargoshasb S, Korayem A, Tabibian SA (2018) voice command detection system for controlling movement of SCOUT robot. In: The 6th RSI International Conference on Robotics and Mechatronics (IcRoM), pp 326–330

3. Bellegarda JR (2014) Spoken language understanding for natural interaction: the Siri experience. In: Natural interaction with robots, knowbots and smartphones, pp 3–14
4. Bourlard HA, Morgan N (2012) Connectionist speech recognition: a hybrid approach, vol 247. Springer Science & Business Media
5. Braun M, Broy N, Pfleging B, Alt F (2019) Visualizing natural language interaction for conversational in-vehicle information systems to minimize driver distraction. J Multimodal User Interfaces 13(2):71–88
6. Buchhop K, Edel L, Kenaan S, Raab U, Böhm P, Isemann D (2017) In-vehicle touchscreen interaction: can a head-down display give a heads-up on obstacles on the road? In: Proceedings of the 9th international conference on automotive user interfaces and interactive vehicular applications, pp 21–30
7. Burnett G, Hazzard A, Crundall E, Crundall D (2017) Altering speed perception through the subliminal adaptation of music within a vehicle. In: Proceedings of the 9th international conference on automotive user interfaces and interactive vehicular applications, pp 164–172
8. Castronovo S, Mahr A, Pentcheva M, Müller C (2010) Multimodal dialog in the car: combining speech and turn-and-push dial to control comfort functions. In: Eleventh Annual Conference of the International Speech Communication Association, pp 510–513
9. Diaconu C, Freedman C, Larson P, Zwilling M (2016) Inventors; Microsoft technology licensing, Llc, assignee. US Patent US9,251,214
10. Endsley MR (2016) Designing for situation awareness: an approach to user-centered design. CRC press
11. Fischer P, Nurnberger A (2008) Adaptive and multimodal interaction in the vehicle. In: IEEE international conference on systems, man and cybernetics, pp 1512–1516
12. Green P (1999) The 15-second rule for driver information systems. In: Proceedings of the ITS America Ninth Annual Meeting, pp 1–9
13. Hidden Markov Model Toolkit (HTK) (2015) Speech vision and robotics group of the Cambridge university engineering department
14. Hossan MA, Memon S, Gregory MA (2010) A novel approach for MFCC feature extraction. In: The 4th international conference on signal processing and communication systems, pp 1–5
15. Kalkhoran LS, Tabibian S, Homayounvala E (2020) Improving the accuracy of Persian HMM-based voice command detection system in smart homes based on ontology method. In: 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), pp 1–5
16. Khare A, Sinha A, Bhowmick B, Kumar K, Gosh H, Wattamar S, Kopparapu SK (2009) Multimodal interaction in modern automobiles. In: Multimodal interfaces for automotive applications, pp 1–4
17. Klakow D, Peters J (2002) Testing the correlation of word error rate and perplexity. Speech Comm 38(1–2): 19–28
18. Korayem M, Azargoshasb S, Korayem A, Tabibian S (2021) Design and implementation of the voice command recognition and the sound source localization system for human–robot interaction. Robotica 39(10):1779–1790
19. Kujala T (2013) Browsing the information highway while driving: three in-vehicle touch screen scrolling methods and driver distraction. Pers Ubiquit Comput 17(5):815–823
20. Kujala T, Grahn H (2017) Visual distraction effects of in-car text entry methods: comparing keyboard, handwriting and voice recognition. In: Proceedings of the 9th international conference on automotive user interfaces and interactive vehicular applications, pp 1–10
21. Marcus A (1995) Principles of effective visual communication for graphical user interface design. In: Readings in human–computer interaction. Elsevier, pp 425–441
22. McCallum MC, Campbell JL, Richman JB, Brown JL, Wiese E (2004) Speech recognition and in-vehicle telematics devices: potential reductions in driver distraction. Int J Speech Technol 7(1):25–33
23. Miller R (2004) User interface design and implementation. Lecture notes. Massachusetts institute of technology
24. Naseri MM, Tabibian S (2020) Improving the robustness of persian spoken isolated digit recognition based on LSTM. In: 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), pp 1–6
25. Nielsen J (1994) Enhancing the explanatory power of usability heuristics. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp 152–158
26. Pfleging B, Schneegass S, Schmidt A (2012) Multimodal interaction in the car: combining speech and gestures on the steering wheel. In: Proceedings of the 4th international conference on automotive user interfaces and interactive vehicular applications, pp 155–162
27. Roider F, Rümelin S, Pfleging B, Gross T (2019) Investigating the effects of modality switches on driver distraction and interaction efficiency in the car. J Multimodal User Interfaces 13(2):89–97
28. Sameti H, Veisi H, Bahrani M, Babaali B, Hosseinzadeh K (2008) Nevisa, a persian continuous speech recognition system. In: Computer society of Iran computer conference, pp 485–492
29. Standardization IOf (2018) ISO 9241-11: 2018, ergonomics of human-system interaction, part 11: usability: definitions and concepts

30. Tabibian S (2017) A voice command detection system for aerospace applications. Int J Speech Technol 20(4):1049–1061
31. Tabibian S (2018) Design and collection of Persian spoken digits based on cell phone. In: Proceedings of the 4th conference on signal processing and intelligent systems, Tehran, pp 1–5
32. Tsimhoni O, Green P (2001) Visual demand of driving and the execution of display-intensive in-vehicle tasks. In: Proceedings of the human factors and ergonomics society annual meeting, vol 23. SAGE Publications Sage CA, pp 1586–1590
33. Veisi H, Haji Mani A (2020) Persian speech recognition using deep learning. Int J Speech Technol 23(4): 893–905
34. Wickens CD, Gordon SE, Liu Y (2003) An introduction to human factors engineering, 2nd edn. Pearson
35. Yang S, Pan Y (2014) A study on methods of multimodal interaction in vehicle based on wheel gestures and voices. In: International conference on human-computer interaction, pp 484–489
36. Zhao D, Wang C, Liu Y, Liu T (2019) Implementation and evaluation of touch and gesture interaction modalities for in-vehicle infotainment systems. In: International conference on image and graphics, pp 384–394

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.