sensors

MDPI

Article

Single-Equipment with Multiple-Application for an Automated Robot-Car Control System

Saleem Ullah [1,†], Zain Mumtaz [1,†], Shuo Liu [2], Mohammad Abubaqr [1], Athar Mahboob [1,3] and Hamza Ahmad Madni [2,3,*]

[1] Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan; saleem.ullah@kfueit.edu.pk (S.U.); zainmumtaz007@gmail.com (Z.M.); mohammadabubaqr@outlook.com (M.A.); athar@kfueit.edu.pk (A.M.)

[2] State Key Laboratory of Millimeter Waves, Department of Radio Engineering, Southeast University, Nanjing 210096, China; liushuo.china@seu.edu.cn

[3] Department of Computer Engineering, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan

* Correspondence: 101101770@seu.edu.cn; Tel.: +86-1782-6500-615 or +92-313-4820-856

† These authors contributed equally to this work.

Received: 17 January 2019; Accepted: 2 February 2019; Published: 6 February 2019

Abstract: The integration of greater functionalities into vehicles increases the complexity of car-controlling. Many research efforts are dedicated to designing car-controlling systems that allow users to instruct the car just to show it what it should do; however, for non-expert users, controlling the car with a remote or a switch is complicated. So, keeping this in mind, this paper presents an Arduino based car-controlling system that no longer requires manual control of the cars. Two main contributions are presented in this work. Firstly, we show that the car can be controlled with hand-gestures, according to the movement and position of the hand. The hand-gesture system works with an Arduino Nano, accelerometer, and radio-frequency (RF) transmitter. The accelerometer (attached with the hand-glove) senses the acceleration forces that are produced by the hand movement, and it will transfer the data to the Arduino Nano that is placed on hand glove. After receiving the data, Arduino Nano will convert it into different angle values in ranges of 0–450° and send the data to the RF receiver of the Arduino Uno, which is placed on the car through the RF transmitter. Secondly, the proposed car system is to be controlled by an android based mobile-application with different modes (e.g., touch buttons mode, voice recognition mode). The mobile-application system is the extension of the hand-gesture system with the addition of Bluetooth module. In this case, whenever the user presses any of the touch buttons in the application, and/or gives voice commands, the corresponding signal is sent to the Arduino Uno. After receiving the signal, Arduino will check this against its predefined instructions for moving forward, backward, left, right, and brake; then it will send the command to the motor module to move the car in the corresponding direction. In addition, an automatic obstacle detection system is introduced to improve the safety measurements to avoid any hazards with the help of sensors placed at the front of the car. The proposed systems are designed as a lab-scale prototype to experimentally validate the efficiency, accuracy, and affordability of the systems. The experimental results prove that the proposed work has all in one capability (hand-gesture, touch buttons and voice-recognition with mobile-application, obstacle detection), is very easy to use, and can be easily assembled in a simple hardware circuit. We remark that the proposed systems can be implemented under real conditions at large-scale in the future, which will be useful in automobiles and robotics applications.

Keywords: android; arduino; bluetooth; hand-gesture recognition; low cost; open source; sensors; smart cars; speech recognition

1. Introduction

A robot is an intelligent machine controlled by a computer application to perform various operations and services. Robots play an essential role in automation across all sectors like manufacturing, military, construction, and medical [1] etc. Robots can not only help humans save time but also increase productivity [2], efficiency, reliability, reduce the use of resources to save energy, and reduce the running cost etc. Meanwhile, robots play a significant role in providing help in such tasks that cannot be done smoothly by a disabled person, i.e., controlling a car using physical devices has become very successful. There are two categories of robots: autonomous robots (edge sensing robots [3], line sensing [4]) and remote-controlled robots (gesture-controlled robots [5]). Therefore, the employment of gesture-controlled robots is one of the more elegant and aesthetic options termed to catch the human-gesture, which is difficult to understand in machine-learning.

There are many methods to capture gesture. Commonly used methods include a data glove [6], camera [7], infrared waves [8], tactile [9], acoustic [10] and motion technology [11]. These embedded systems [6–11] are designed for particular control and can be optimized to increase reliability, performance, and reduce the cost and size of the device. Moreover, researchers are showing tremendous interest in gesture recognition, building robots and many other devices that are directly controlled by human gestures. The gesture control mechanism is applied in various fields like socially assistive robots, augmented reality, emotion detection from facial expressions, and recognition of sign languages [12–14] etc. Furthermore, the emotional gesture [12] identification from the face is also under investigation. Similarly, with the arrival of the smartphone and other advanced technologies, operating machines have become more adaptable. In this regard, the smartphone communicates with the microcontroller of the robot via radio frequency (RF), Bluetooth or Wi-Fi.

The traditional robot system has been limited to the remote system in which the desired actions can only be done with a unique and specific remote. In this scenario, if the remote is broken or lost, the robot loses control and leads to hazards and is waste of money as well. To overcome this remote-control concept, controlling the robot using a gesture recognition technique with an accelerometer [15–33], Bluetooth android application [34–41], and voice recognition [42–50] have been proposed. Thus, in previous literature, most of them are line-sensing robots [51–53] that are controlled with Infrared LEDs, HF band short-range RFID systems [54], and CNY 70 sensors [55]. In addition, vision-based technology [56,57] is also used to control the movement of the robot. Furthermore, an automated robot control system with a six sense [58–60] and ZigBee based system [61–64] has also been implemented. In addition, another approach has been introduced to allow the robot to avoid obstacles [65–67], in which the robot stops moving or changes its route of motion. Similarly, in edge detector [3] systems, the robot keeps moving due to the detection of an edge; and in an line follower systems [4], the robot follows a unique line and starts moving. A motion capture sensor recognition system [16,22] is also used to implement an accelerometer-based method to interact wirelessly with an industrial robotic arm.

Apart from the traditional robot control system, the term "internet of things" (IoT) [68–74] is also essential for connecting a robot with the internet to allow users to control the robot from anywhere at anytime. These wireless systems are providing vital help to robot self-regulation systems by using Wi-Fi and cloud computing, etc. As far as we know, a need still exists for the design of single-equipment with a multiple-application system that supports various tasks (e.g., hand-gesture recognition, android mobile application control and voice recognition concepts, monitoring the obstacles) and is very easy to use and can be easily assembled in a simple hardware circuit.

In this paper, we introduce the design and experimentally demonstrate that a robot-car can be controlled with hand movement by using the technique of hand-gesture. This work is accomplished with the conventional arrangements of the Arduino microcontroller, accelerometer, RF transmitter/receiver, Android mobile-application, Bluetooth, and motor module. The proposed robot-car is controlled via gesture recognition technique by measuring the angles and position of the hand. In this way, the robot will move according to the movement of the hand. Furthermore, we extend this system to allow the robot to be controlled by just a click on the cellphone with an Android operating system, and voice recognition via Bluetooth technology. There are two main controllers in android the application. The first one is the touch buttons. The robot will move accordingly as the user touches the button. The second one is voice recognition. The robot will follow and move accordingly as the user says the operating command. Most importantly, an obstacle detection sensor is set to detect the obstacle in front of it, and when a sensor detects the obstacle, it stops moving. Hence, the proposed systems of controlling the robot car with both gesture control and android application control are performed and displayed in a lab-scale prototype to confirm that the proposed designs can be easily implemented in large and real-scale conditions in the future.

The innovative element of the proposed work in this paper is an Arduino that is a user-friendly microcontroller that can be easily available on demand. On the other hand, automated systems can be designed with raspberry pie, ZigBee, and other microcontrollers that are costly and complicated for the encapsulation process to assemble the various functionalities in a simple hardware circuit. An accelerometer based hand-glove is also the key element to investigate the car-controlling set-up as a reciprocal of vision-based gesture recognition [75] (the aim is in the field of service robotics to control a robot equipped with a manipulator), and finger-gesture recognition [76] (the aim is to make it feasible to interact with a portable device or a computer through the recognition of finger gestures).

Furthermore, the motivation of conducting this research is to facilitate the old-age and/or physically disabled people who cannot walk or who are facing problems in performing their daily life activities. The proposed work in this paper can be easily implemented to design and operate a wheelchair with the help of hand-gestures and mobile-application systems, which in turn commands the wheelchair. As the wheelchair moves front, back, right, and left, a partially paralyzed patient can move freely. Most importantly, the person will be able to control the wheelchair in a narrow space because this proposed scheme uses sensors that are helpful in avoiding collisions. Similarly, it might be difficult or unsafe for humans to do some tasks like picking up dangerous chemicals. Thus repeated pick and place actions are important in certain industries. In this field it is quite complicated to control the robot or machine with a remote or switches; sometime the operator may get confused between the switches and buttons, so we aim to design a new concept to control the machine by hand movement combined with a Bluetooth android application that will simultaneously control the movement of the robot. Thus, the aim of this proposed research is to construct a multiple-functional robot with low cost that can be used in various activities.

The remaining content of the paper is ordered as follows. In Section 2, the idea of the automatic control robot car is introduced with a detailed explanation of the electronic components that are used in the proposed system, based on gesture recognition with the mobile-based android application. Also, the experimental results of the proposed systems in a lab-scale prototype are presented in Section 3. Section 4 concludes the paper. Section 5 presents the future work.

2. Materials and Methods

For the simplicity of analysis, Figure 1 demonstrates the complete working mechanism and the features of the proposed automatic robot car. Whereas, I/P and O/P represent the flow of the system as input and output. There are two modes of transmission and controlling of the robot car. The first one is the hand-gesture system. The hand accelerometer first senses the acceleration forces from the direction of the hand and sends them to the Arduino Nano that is attached to the hand. After receiving the data, Arduino Nano converts it into different angles, between 0–450°, and sends it to the RF receiver of the Arduino Uno that is attached to the robot car through the RF sender.

After receiving the data, the Arduino Uno of the car will measure the received angles with the predefined set of angles and send a signal to the motor module to move the wheels of the robot car accordingly to the angles. It can be noticed that the range of angles are predefined for the wheel movement of the robot car to move forward, backward, brake, left and right.

The second mode is to control a robot car with an android mobile application, which is a built in android mobile application available at the google play store that can be easily downloaded [77].

In this system, when the user presses the corresponding touch button, a signal is transferred to the Arduino UNO that is attached to the car through the built-in mobile Bluetooth device. After receiving the following signal command, Arduino will check this signal with a predefined instruction that is programmed via coding and send the following signal to the motor module to move the wheels of the robot car accordingly to the received signal.

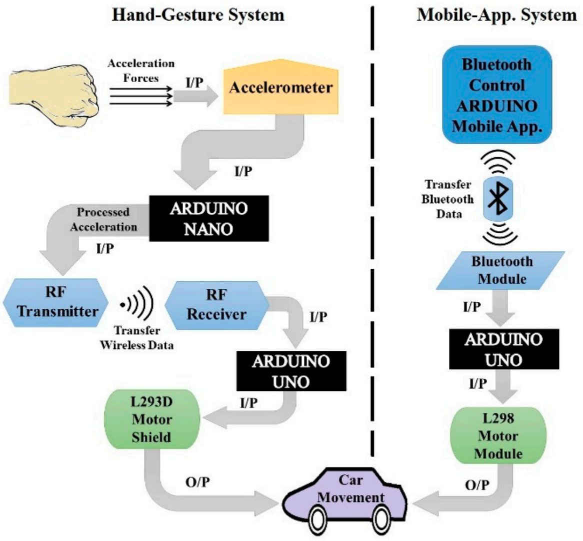

Figure 1. The architecture design of car controlling with hand-gesture and mobile application system.

2.1. Electronic Components

Various electronic components are used for creating electronic circuits. Consequently, our proposed circuit diagrams also contain those components that are specified in Table 1.

Table 1. Specification of electronic components used in to design the proposed system.

Components	Specifications
Arduino UNO [50,78]	28 pins; Operating voltage: 7–12 V
Arduino Nano [79]	30 pins; Operating voltage: 7–12 V
MPU6050 Accelerometer [80]	8 pins; Operating voltage 3.3 V
RF Sender/Receiver [81]	Sender (3 pins; Operating voltage 3–12 V; Transmission range: 90 m), Receiver (4 pins, Operating voltage 5 VDC)
L293D Motor Shield [82]	Supply-Voltage Range: 4.5–36 V; Output current: 600 mA/channel
Bluetooth Module HC-05 [83]	6 pins; Operating voltage: 3.3–5 V; Transmission range: 100 m
Android Mobile Application [77]	Android compatible
L298 Motor Module [84]	Operating voltage: 5 V; Max power: 25 W

2.1.1. Arduino UNO

The Arduino Uno [50,78] is a microcontroller board that is principally based on the ATmega328 microcontroller series and has an integrated development environment (IDE) to write, compile, and upload the programming codes to the microcontroller. Various sensors will forward the environmental data as an input to the microcontroller, which will correspondingly send the attached peripherals, e.g., actuators. It has a total number of 28 pins; 14 digital input/output pins (six are pulse width modulation (PWM) pins) and six pins are the analogs used for interaction with the electronic components like sensors, motors; 3 GND pins (for grounding) and remaining pins for 3.3 V, 5 V, VIN, RESET and AREF (Analogue Reference). Arduino contains a microcontroller with a 32 KB storage memory, 2 KB of SRAM (Static random-access memory), and 1 KB of EEPROM (Electrically Erasable Programmable Read-Only Memory). Arduino primarily supports a C programming language compiler, macro-assemblers, and evaluation kits. It also has 16 MHz ceramic resonators, a USB connection jack for connecting with a computer, a jack for external power supply, an ICSP (in-circuit serial programmer) header, and a reset button to reset to the factory settings. It operates with an input voltage of 7 to 12 V (limit up to 20 V).

2.1.2. Arduino Nano

Arduino [79] is comprised of a microcontroller of ATmega328P with 32 KB memory, of which 2 KB is used for the bootloader, 2 KB of Static random-access memory (SRAM), 16 MHZ Clock Speed, and 1 KB of electrically erasable programmable read only memory (EEPROM). ATmega328P supports C compiler, evaluation kits, and macro-assemblers. It has a total 30 pins of which 14 are digital input and output pins (six are pulse width modulation (PWM)), 8 Analog IN pins, 2 GND pins, 2 RESET and remaining pins for VIN, 5V, 3V and REF respectively. It has a USB connection jack, an external power supply jack, an ICSP header and a reset button. It operates with an input voltage of 7 to 12 V (limit up to 20 V).

2.1.3. MPU-6050 Accelerometer

The MPU-6050 sensor [80] holds a MEMS (Micro-Electro-Mechanical Systems) accelerometer and a MEMS gyroscope only in the single chip. It contains 16-bits analogue to digital conversion hardware for each of the channels and it captures the x, y, and z channel at the same time. MPU-6050 interfaces with the Arduino microcontroller using the I2C protocol. It has a total number of eight pins named as VCC (for provides power), ground (for grounding of system), serial clock (SCL used for providing clock pulse for I2C communication), serial data (SDA used for transferring data through I2C communication), auxiliary serial data (XDA is optional but can be used to interface other I2C modules with MPU6050), auxiliary serial clock (XCL is also optional but can be used to interface other I2C modules with MPU6050), AD0 (If more than one MPU6050 is used a single MCU, then this pin can be used to vary the address). Interrupt pin (INT) indicates that data is available for MCU to read, and operating voltage is 3–5 V.

2.1.4. RF Transmitter and Receiver

RF modules are 433 MHz RF transmitter and receiver modules [81] that are used to transmit and receive the infrared waves. RF transmitter consists of three pins; ATAD (Signal pin used to send data for the receiver), VCC (for providing voltage) and Ground pin (for grounding the module). Its working voltage is 3–12 V and consumes 12 V power. The RF transmitter can transmit up to 90 m in an open area. The RF Receiver module consists of four pins; VCC (for providing Voltage), two DATA pins for receiving data from transmitter and Ground (for grounding the module). Its working voltage is 5 VDC.

2.1.5. L293D

The L293D [82] is quadruple high-current half-H drivers that are used to control the direction and speed of up to four direct current (DC) motors with individual 8-bit speed selection simultaneously up to 0.6 A each. It is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 to 36 V. It has eight output pins for connecting four DC motors or two Stepper motors, one reset button, six pins for connecting two servo motors, one +M pin for providing external motor power, five Analog input pins, thirteen digital input pins for connecting with Arduino and ground pins for grounding.

2.1.6. Bluetooth Module HC-05

The HC-05 [83] is a Bluetooth SPP (Serial Port Protocol) module that is designed for direct wireless serial connection setup and can be used in a Slave or Master configuration, giving it an outstanding solution for wireless transmission. This serial port Bluetooth module is entirely adequate Bluetooth V2.0 + EDR (Enhanced Data Rate) 3 Mbps Modulation with complete 2.4 GHz radio transceiver and baseband. It has a total number of six pins; an ENABLE pin to toggle between the Data Mode and AT command mode, a VCC pin for providing voltage, a Ground pin for grounding, a TX-Transmitter for transmitting serial data, a RX-Receiver for receiving the serial data, and a State pin (to check if Bluetooth is working correctly or paired/unpaired). Its input voltage is 3.3–5 V and can transmit up to 90 m.

2.1.7. Android Mobile Application

An Android application is a software application (developed with a computer programming language) which run on the Android platform. The application for controlling a robot car is available at [77] and easily downloadable.

2.1.8. L298N Motor Module

An L298N motor module [84] is a heavy-duty dual H-bridge controller, which is used to control the direction and speed of single or two direct current (DC) motors of up to 2 A each, having a voltage between 5 V to 35 V. It has principally four output pins for the connection of the DC motors, four input pins to receive the signal from the microcontroller, two enable jumpers (remove one of the corresponding jumpers and connect to the pulse width modulation pins to control the speed of DC motors). It also has an onboard 5 V regulator that removes that regulator if the supply voltage is up to 12 V.

3. Designing Methodology

3.1. Hand-Gestures Recognition

Figure 2 shows the circuit design of hand-gestures, which control the robotic car using hand movement. In this scenario, the robotic car will move in the same direction as the direction of hand rotation. Figure 2a is for the hand-gesture setup and Figure 2b is for the car setup. First, we describe the hardware implementation of the hand-gesture setup. In this task, one Arduino Nano, one MPU-6050 accelerometer and one RF transmitter were used. The SLC pin of the MPU-6050 module is connected to the Arduino analogue Pin A5, SDA to the A4, Ground pin to the GND and VCC (voltage at the common collector) pin to the 5 V pin of the Arduino. DATA pin of the RX-transmitter is connected to the Arduino digital PIN D8, VCC pin to 5 V and ground pin to GND port, as shown in Figure 2a.

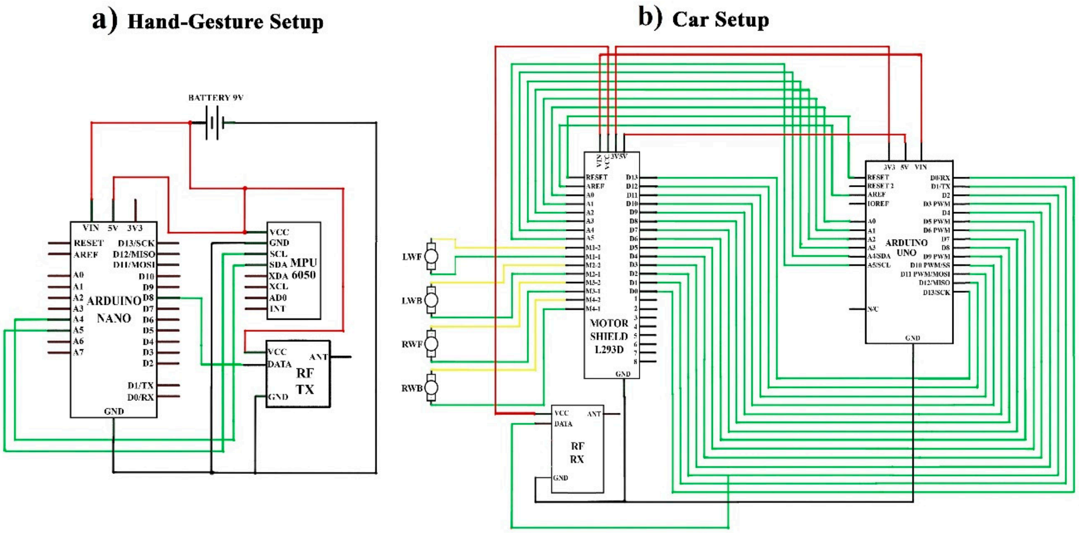
a) Hand-Gesture Setup
BATTERY 9V
VIN 5V 3V3
RESET AREF A0 A1 A2 A3 A4 A5 A6 A7
D13/SCK D12/MISO D11/MOSI D10 D9 D8 D7 D6 D5 D4 D3 D2 D1/TX D0/RX
ARDUINO NANO
GND
VCC GND SCL SDA XDA XCL AD0 INT
MPU 6050
VCC DATA GND ANT
RF TX
b) Car Setup
LWF
LWB
RWF
RWB
VIN VCC 3V5V
RESET AREF A0 A1 A2 A3 A4 A5 M1-2 M1-1 M2-2 M2-1 M3-2 M3-1 M4-2 M4-1
D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 1 2 3 4 5 6 7 8
MOTOR SHIELD L293D
GND
3V3 5V VIN
RESET RESET 2 AREF IOREF A0 A1 A2 A3 A4/SDA A5/SCL N/C
D0/RX D1/TX D2 D3 PWM D4 D5 PWM D6 PWM D7 D8 D9 PWM D10 PWM/SS D11 PWM/MOSI D12/MISO D13/SCK
ARDUINO UNO
GND
VCC DATA GND ANT
RF RX

Figure 2. Circuit design of the car controlling system using hand gestures recognition. (**a**) Shows the schematic of the hand-gesture setup. (**b**) Displays the schematic design of car setup.

After hardware implementing of the hand gesture, we move to the implementation of car setup. In this task, one L293D motor module, one Arduino Uno, and one RF-receiver were used. Digital pins from A0–A13 of L293D motor module are connected to the digital pins A0–A13 and Analog pins A0–A5 are connected to the analog pins A0–A5, and similarly, Vin pin to Vin port, 3 V pin to 3 V port, 5 V pin to 5 V port, reset pin to reset port, ground pin to GND port and AREF pin to the AREF port of Arduino Uno as displayed in Figure 2b. Further, M1.1–M4.1 pins are connected to the negative terminal and M1.2–M4.2 pins of the motor module are connected to the positive terminals of the motor. Similarly, a VCC pin of the RF-receiver is connected to the VCC pin of the motor module, the DATA pin to the digital pin D2 of the motor module and Arduino and at last, the Ground pin to the GND port of the Arduino Uno. The complete software code of this case is presented in Figure S1 of the Supplementary Materials.

3.1.1. Movement of Motors with Hand-Gesture

As the user moves their hand, the reading of the accelerometer will change, and then it will be recaptured by the application. There are genuinely two values: One is the minimum value (X_{range}), and another is the maximum value (Y_{range}), and the range is defined using these two values for each function of the car. In simple words, the set of ranges are defined for the movement of the robot-car in a specific way. If the received data by the application lies within these specified values, then the corresponding decision will be made. This decision value will be sent to the microcontroller, which then processes it to understand the corresponding gesture. After knowing the gesture, it will send a signal to move the robot car accordingly. There is a total of four DC motors, one motor for each wheel (two for left diagonal wheels, and two for right diagonal wheels) used in the construction of this car. The motors are controlled by the L293D motor shield.

Figure 3 represents the main idea of gesture recognition and car movement. When the user tilts his hand downward, the gesture is recognized as the forward movement, all four wheels of the motors will rotate forward and the robot will move in the forward direction, which can be easily seen in Figure 3a. Figure 3b illustrates the case, when the user tilts his hand upwards, the gesture is recognized as the move backward gesture, and all the four wheels of motors will rotate backward, and the robot moves in a backward direction. In Figure 3c, the hand is in the straight position, and if the gesture is recognized to stop the car, all four wheels will stop moving. When the user tilts his hand in the right direction, the gesture is recognized as the right turn, so the right diagonal motors (top right and back right motors) will rotate forwards, and the robot moves in the right direction as shown in Figure 3d. Similarly, when the user tilts his hand in the left direction, the gesture is recognized as the left turn, so only left diagonal motors (top left and back left motors) will rotate forward and the robot moves in the left direction, as represented in Figure 3e.

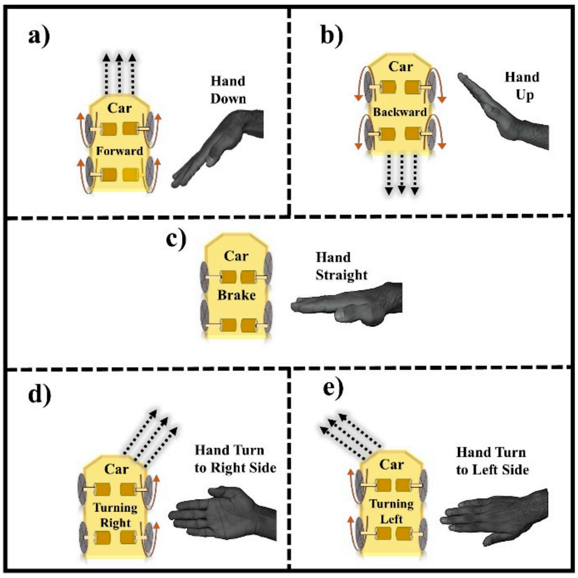

Figure 3. The theme of gesture recognition (i.e., movement of hand, motors, and wheels): (**a**) Hand Down; (**b**) Hand Up; (**c**) Hand Straight; (**d**) Hand Turn to Right Side; (**e**) Hand Turn to Left Side.

It is worth noting that the values of the tilt of the user's hand determine the right or left turn is a cutting turn. The cutting turn is the one in which the car changes its direction without slowing down speed. If it is a right turn, the car will start moving in the right circle direction, or if it is left turn, the car will move in the left circle direction. The hand of a user will never lie within the two gestures (the values for each action is defined differently from another action), i.e., the value of the accelerometer will not lie in the threshold values of two directions (left turn and forward, left turn and backward, right turn and forward, right turn and backward).

3.1.2. Results and Discussions

In the beginning, the accelerometer (attached with the hand glove) will sense the acceleration forces that are produced with the hand movement at that time, and it will consequently transfer the data to the Arduino Nano that is placed on the hand glove. After receiving the data, the Arduino Nano will convert it into different angle values of ranges 0–450° and send the data to the RF Receiver of the Arduino Uno that is placed on the car through the RF Transmitter, as is clearly exhibited in Figure 4.

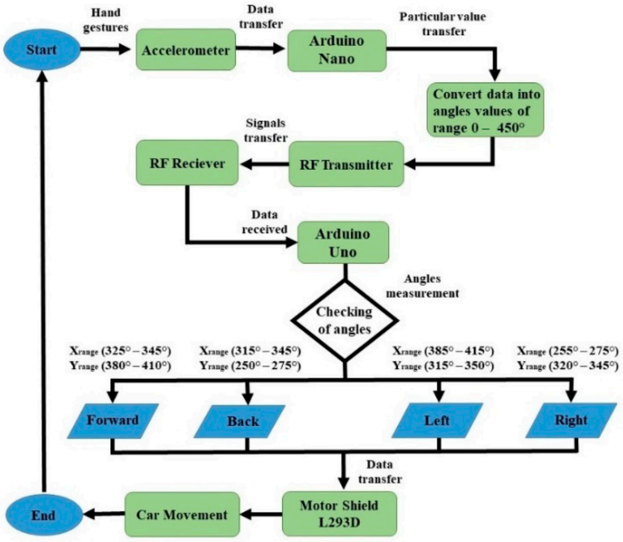
Start
Hand gestures
Accelerometer
Data transfer
Arduino Nano
Particular value transfer
Convert data into angles values of range 0 – 450°
Signals transfer
RF Reciever
RF Transmitter
Data received
Arduino Uno
Angles measurement
Checking of angles
X_{range} (325° – 345°) Y_{range} (380° – 410°)
X_{range} (315° – 345°) Y_{range} (250° – 275°)
X_{range} (385° – 415°) Y_{range} (315° – 350°)
X_{range} (255° – 275°) Y_{range} (320° – 345°)
Forward
Back
Left
Right
Data transfer
Motor Shield L293D
Car Movement
End

Figure 4. The flow diagram of the hand-gesture based car controlling.

When Arduino Uno receives these angles, it will measure these with the predefined set of angle values. Arduino Uno will check if X_{range} lies between 325–345° and Y_{range} lies between 380–410°, it will send a forward signal to motor shield. If Arduino Uno founds X_{range} in between 315–345° and Y_{range} in between 250–275°, it will send a backward signal to the motor shield. Similarly, if X_{range} lies between 385–415° and Y_{range} between 315–350°, the Arduino will send a Left signal to the motor shield. Furthermore, if X_{range} lies within 255–275° and Y_{range} lies within 320–345°, Arduino will send a right signal to the motor shield. Thus, after the obtained signal, the motor shield will control the car's movement.

Initially, the robot car will not move and stay motionless when it does not receive any signal from an accelerometer or when the accelerometer of hand glove is in the straight position. Whenever the user tilts or rotates his hand palm glove to any direction, the angles will be measured of that direction by an Arduino and a command will be sent to the motor shield to turn ON the corresponding motors and the robot car will start moving in a similar to the hand palm direction.

Figure 5 shows the final presentation of the proposed robot car system that controlled it via a hand gesture recognition using Arduino. Figure 5a represents the complete hardware implementation of the hand glove and the robot car. Figure 5b shows that the robot car is moving to the forward direction (all the four wheels is moving) because the hand is bent in the downward position. In Figure 5c, the robot car is moving in a backward direction as the hand's position is upwards. In Figure 5d, the robot car is not moving because the hand is in the straight position as a brake command. In Figure 5e, the robot car is moving in the right direction because the hand is turned in the right position. Similarly, in Figure 5f, the car is moving to the left direction because the hand is turned in the left position.

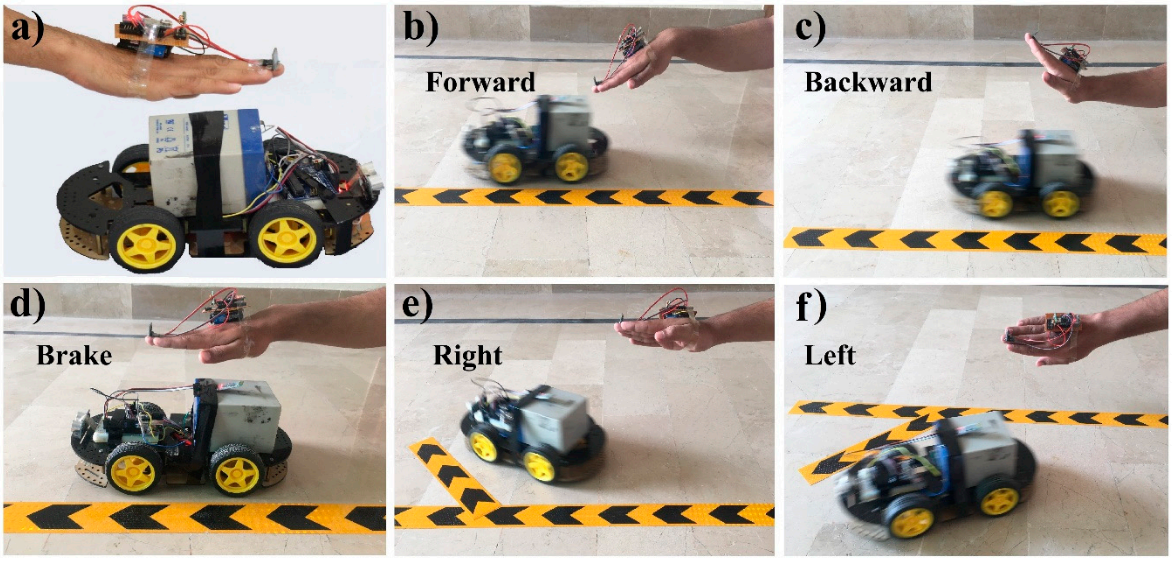

Figure 5. Result diagrams of the automatic robot car controlling system using hand-gestures. (**a**) Hand gesture and car's hardware are illustrated (**b**) The hand is tilted in a downward position, so the car is moving forward. (**c**) The hand is tilted in the upward position, so the car is moving backward. (**d**) The car stopped when the hand is in a straight position. (**e**) The car is moving to right direction as the hand-tilted to the right side. (**f**) The hand is tilted to the left side, so the car is moving to left direction. The complete demonstration video of the proposed system is available online [85].

3.2. Mobile Application System

According to the motive, the primary idea of this paper is to produce an innovation for the robot car that will allow it to be controlled by hand-gestures as well as an Android mobile application. As presented in Figure 4, the robot car is controlled via hand-gestures. Here, we enhanced our task by controlling the robot car based on the Android mobile application (e.g., touch button control and voice recognition), and the circuit design of this mode in a lab-scale model can be seen in Figure 6.

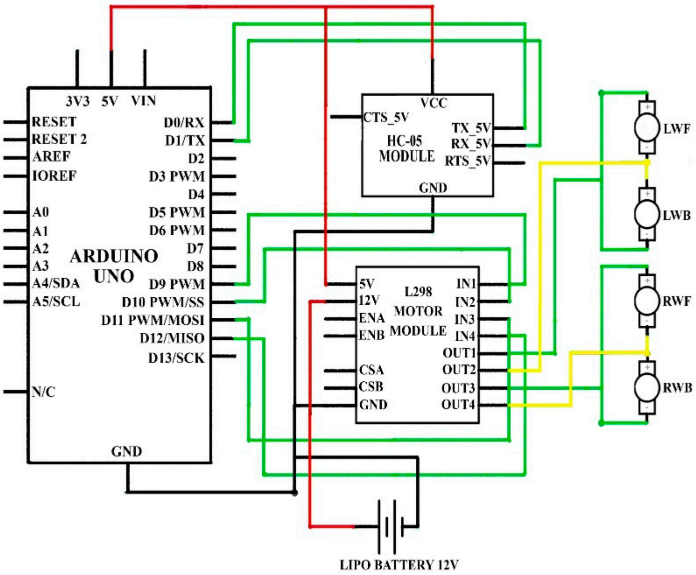

Figure 6. Circuit design of mobile-application system to control car.

In this task, an Arduino UNO, one L298 motor module, one HC-05 Bluetooth module, one 12 V battery and four DC motors are used. Hence, the TX, RX pin of the Bluetooth module is connected to pin number D0, D1, VCC pin to 5 V port and ground pin to ground port of the Arduino UNO. Similarly, the 5 V pin and ground pin of the motor module is connected to 5 V port and ground port of Arduino UNO. The 12 V pin of the motor module is connected to a positive terminal of a 12 V battery and the negative terminal of the battery is connected to the ground port of Arduino, motor module, and Bluetooth module.

Moreover, IN1–IN4 pin of the motor module is attached to Arduino PIN D9–D12 respectively. OUT1 pin of the motor module is connected to the positive terminal of the LWF motor (left wheel forward) and the negative terminal of the LWB motor (left wheel backward), OUT2 pin to the negative terminal of LWF motor and positive terminal LWB motor, OUT3 pin to positive terminal of the RWF motor (right wheel forward) and negative to the RWB motor (right wheel backward). Likewise, OUT4 pin to the negative terminal of the RWF motor and positive terminal of RWB motor as exhibited in Figure 6. The complete software code of this case is presented in Figure S2 of the Supplementary Materials.

Results and Discussions

Figure 7 shows the flow diagram to control the robot car with an Android-based mobile application. In this scenario, the robot car will move with the built-in Android application with different methods like touch button control (arrow and simple buttons), voice recognition. Firstly, with the touch buttons; after establishing the connection to the Android mobile application with Arduino through the Bluetooth module, whenever the user presses any of the touch buttons in the application, the corresponding signal is sent to the Arduino Uno. After receiving the signal, Arduino will check this with predefined instruction for moving forward, backward, left, right, and brake, then send the command to the motor module to move the robot in the corresponding direction.

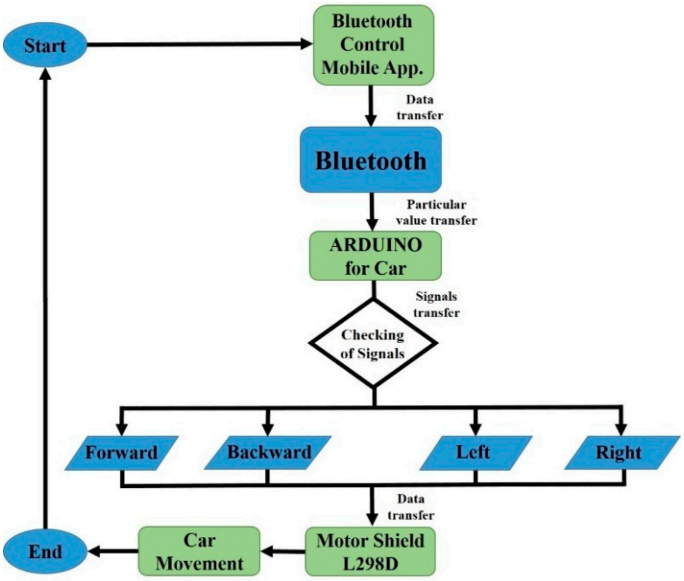

Figure 7. The flow diagram to control robot car with mobile-application.

The second method to control the robot car is voice recognition through the GUI of the android application. When the user selects the voice recognition option in the application and speaks the specific word (left, right, forward, backward and brake), the application sends this voice command signal to Arduino, and it will begin measuring the signal and complete the operation as it is defined in the previous method to control the robot car with touch buttons.

Figure 8 shows the result diagrams of the proposed robot car system controlled via the touch buttons of a Bluetooth android mobile application using Arduino. In this way, Figure 8a represents the complete working interfaces, where Figure 8a i is the starting screen of the mobile application; Figure 8a ii presents different controlling interfaces like touch arrow buttons, accelerometer, simple touch buttons, and voice recognition etc. Figure 8a iii shows the four touch arrow buttons; and Figure 8a iv illustrates the six simple buttons (functionality for each button is predefined such as, 1 for forward, 2 for backward, 3 for left, 4 for the right, 5 for brake and six is undefined).

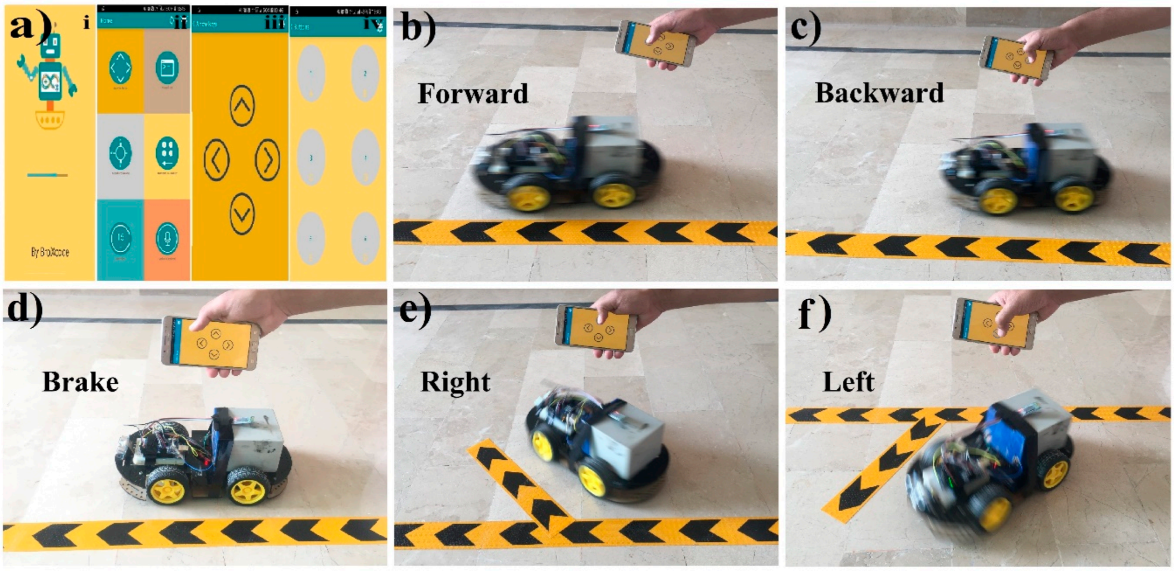

Figure 8. Result diagrams of the automatic robot car controlled with touch buttons of mobile-application. (**a**) i. Starting interface of the android application. ii. The homepage of an android application showing six different options of car control. iii. Touch arrow buttons interface for controlling car. iv. Simple touch buttons interface for controlling car. (**b**) The forward arrow button is touched, so the car is moving forward. (**c**) The backward arrow button is touched, so the car reverses. (**d**) The car is not moving (stopped) because none of the arrow buttons were touched. (**e**) The car is moving to the right direction as the right arrow button was touched. (**f**) The left arrow button was touched, so the car moves to the left direction. The complete demonstration video of the proposed system is available online [85].

Although, there are five different options to control the car, here, as an example, we used touch-buttons option to demonstrate the working functionality of the proposed system. In Figure 8b, the robot car is moving forward after because the forward touch arrow button is touched in the mobile application. Same as in Figure 8c, the car is moving backward because the backward touch arrow button is pressed. In Figure 8d, the car is not moving and is stopped at mid-point because none of the touch arrow buttons were pressed. In Figure 8e, the right touch arrow button is touched in the mobile application and the robot car is turned to right direction. Similarly, in Figure 8f, the robot car moves in the left direction after pressing the left arrow button.

Furthermore, mobile-application for voice recognition is utilized. The voice recognition module in the system is limited mainly to the five voice commands (forward, stop, backward, turn left, turn right). This application converts the voice command to text and sends it to the robot via Bluetooth. The text is sent to the microcontroller from the mobile phone and it will be compared with the preprogrammed command such as left, right, etc. The microcontroller gives the commanding signal to both the wheels through the motor driver when the user gives the left command, then it stops the left wheel and the right wheel will move, moving the robot in the right direction. This android application uses google speech [86] to text to recognize and process a human voice. The method to operate the voice-controlled system is as follows: Firstly, download and install the app on the mobile phone. Turn on the mobile's Bluetooth and run the application. Here, make sure that the robot-car is switched on, select connect to the HC-05 to authenticate the pairing (pairing password is 1234 or 0000). Finally, click on the microphone symbol on the voice recognition interface and give the voice commands. The given command will convert it into text. As the phone is connected to the microcontroller using a Bluetooth module. After the conversation of the voice command into the text, the app will send the necessary data to the microcontroller using the Bluetooth of the phone. According to the command, the robot will move forward, backward, left, right or stop.

Figure 9 shows the result diagrams of the proposed robot car system that are controlled via voice recognition through a Bluetooth mobile-application. In this process, Figure 9a represents the voice recognition interface where Figure 9a i is the primary interface for voice recognition, and Figure 9a ii shows the interface to input voice after pressing the voice button.

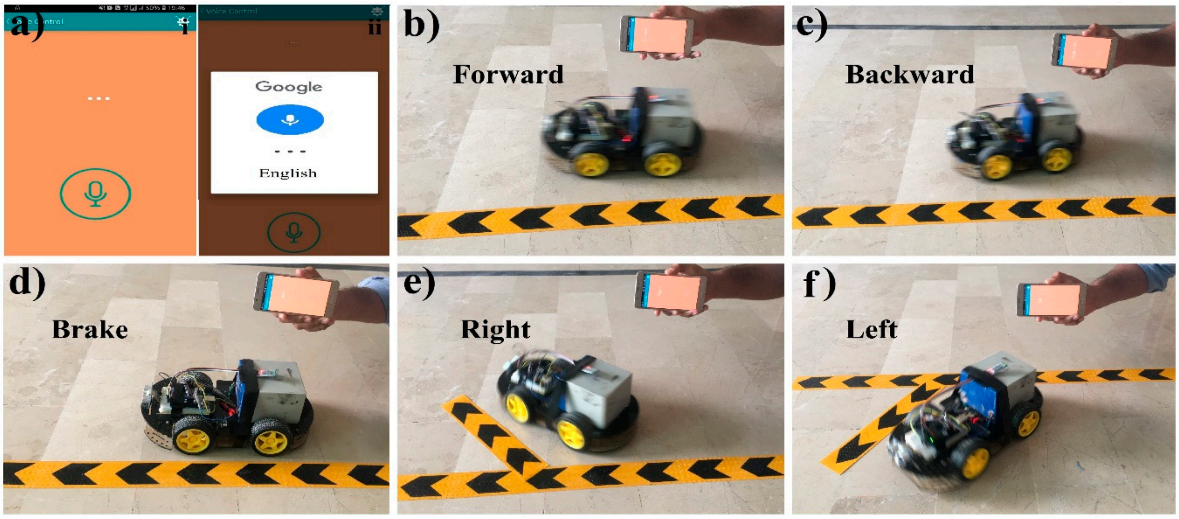

Figure 9. Result diagrams of the automatic robot car controlled with voice recognition in the mobile-application. (**a**) i. Starting interface of the voice recognition android application. ii. Voice is recognized after pressing the voice button of the android application. (**b**) The voice is recognized as a forward command, so the car is moving forward. (**c**) The voice is identified as a backward command, so the car is moving reversely. (**d**) The car is not moving (stopped) because the voice is recognized as a brake. (**e**) The car is moving to the right direction as the voice is identified as a right command. (**f**) The voice is recognized as a left command, so the car is moving to the left direction. The complete demonstration video of the proposed system is available online [85].

In Figure 9b, the robot car moves in a forward direction after voice recognition because the user inputs the forward move voice command to the mobile application. Similarly, the car is moving in a backward direction after voice recognition because the user inputs the backward move voice command as seen in Figure 9c. Meanwhile in Figure 9d, the user inputs the brake voice command to the application, the robot car is not moving. On the other hand, in Figure 9e, the robot car is moving in the right direction after voice recognition because the user inputs the right move voice command. Similarly, the robot car is moving in the left direction after voice recognition because the user inputs the left move voice command, which can be viewed in Figure 9f.

We used the Arduino-based control system with wireless connectivity, in which the robot car could be automatically controlled based on hand gesture recognition and Android application (touch buttons and voice recognition), to avoid any limitations of the specifications, wireless range, etc. In some cases, we required wireless connectivity (WIFI module) to make the system more scalable for the easy integration of new devices. The Wi-Fi module could be replaced with the Bluetooth module to extend the wireless range for this system further. Moreover, the reported systems are presented as lab-scale prototypes, as aforementioned; therefore, the obstacle detection feature can be implemented for the real-scale facilities to improve safety measurements. Similarly, the proposed systems can be managed for real-scale facilities by applying other technologies like a camera [87].

It is worth noting we should consider the cases when the robot car collides with any obstacle in front of it while moving with hand gestures or mobile-application. Thus, for improving the safety measurements, we proposed a system to avoid the robot car from a collision with an obstacle by the help of a sensor (the car will stop 20 cm before the obstacle) as illustrated in Figure 10. In Figure 10a, the car is not moving to the forward direction whereas the hand is bent in the downward direction and Arduino recognized it as a forward command. Similarly, the car is not moving in a forward direction although the user pressed the forward arrow button in a mobile-app because the sensor senses that there is an obstacle in front of the car, as illustrated in Figure 10b.

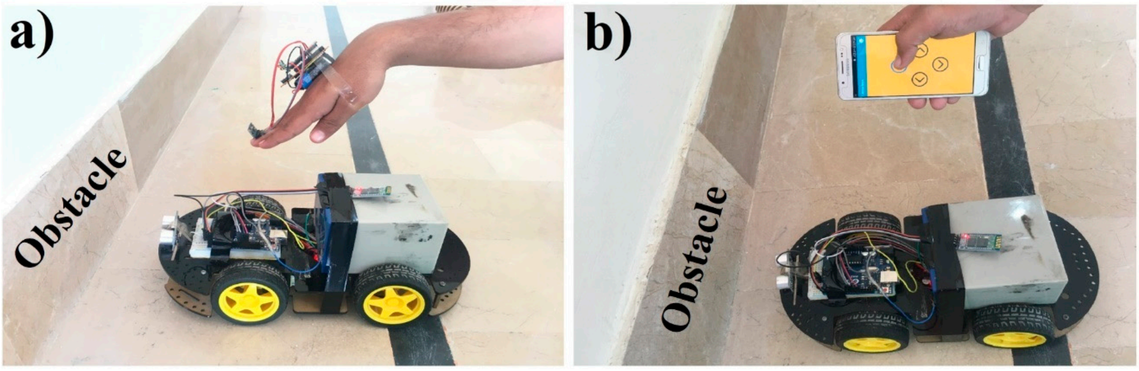

Figure 10. Avoidance of robot car from obstacles using hand-gestures (**a**) and mobile-application (**b**).

The effectiveness of the proposed gesture-recognition technology will routinely add yet another dimension to human communications with consumer electronic appliances such as computers, media tablets and smartphones, video-game consoles, Microsoft's Xbox, and PlayStation etc. The proposed gesture control and mobile-application technique is much more effective for different purposes; in the medical field it can be used by disabled patients to send messages by making a subtle gesture. It can also control electronic devices such as LEDs, fans, TVs, and air conditioners. It can help by giving feedback to augmented reality devices. It can also be paired with a smartphone to provide a more interactive and effective platform to control and interact with it, and it can also be used to control a modern electric vehicle via pairing it to the master control of that vehicle using Human Machine-Interface (HMI) technology.

4. Conclusions

In this paper, a design scheme for a single-equipment with multiple-application system for controlling a robot car based on Arduino has been demonstrated, which can be programmed to react to events (based on hand gesture recognition and an android mobile application as described above) and to cause corresponding actions. The proposed system is presented with principally two operational modes in which the first system uses a mechanism for controlling the robot car based on hand gesture recognition (car moves similarly to the direction and position of hand). This system is further expanded upon to compose the second mode that controls the robot car based on a mobile-application with touch buttons and voice recognition. The proposed systems have the capability to detect obstacles in front of the car. The hardware implementations of the introduced systems were provided at a lab-scale prototype to prove the simplicity, adaptability, dependability, specificity, and real low-cost of the system. We affirm that the introduced systems could be easily implemented under real requirements at large-scale in the future, and they can be efficiently implemented in smart cars, personal home robots, articulated robots, parallel robots, humanoid robots, quadrupedal robots, and hybrid robots etc.

Meanwhile, the proposed single-equipment with a multiple-application system has advantages because it is user-friendly, low-cost, low-power consumption, simple and easy to use, and the system is smaller in size. Thus a small space is required for it to adjust to the hardware circuits. In addition, the designed prototype is highly robust against unexpected problems, and can be easily extended further in the hardware section and various applications can be added to reduce the human effort of upgrading. Similarly, voice commands are sent and received through wireless serial transmission with the help of Bluetooth technology. On the other hand, as the range of Bluetooth technology is up to 10–15 m, the distance of processing the proposed system is smaller. The delay in transmission and response commands becomes high if the Bluetooth connection gets dropped frequently. Moreover, if there is noise or some other sound in the surroundings, the number of errors will increase in the proposed voice-recognition scheme. A limited number of gestures are presented here, but the algorithm can be extended in several ways to recognize a more comprehensive set of gestures.

5. Future Work

Future work will build upon the improvement of the recognition system in order to increase accuracy and more gesture methods. GPS can be added to the proposed automated system by the help of which its location can be tracked. A gesture-controlled system can be controlled by head motion, eye blink, and voice-activated power. In addition, the power supply provided to the receiver can be provided through a solar power system as supply provided by solar power is in the form of a direct current (DC) and a receiver also works on DC. It can be implemented with other robot controllers without significant change. By utilizing Wi-Fi, the processing range can be extended by installing routers on short distances and by using a GSM module for wireless transmission. An onboard wireless camera can be installed, which will provide live streaming and can be used for controlling the car from faraway places. The accuracy of this system for dynamic automobiles can be improved by assessing the driving conditions [88] using a low-cost data acquisition system [89].

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/19/3/662/s1, Figure S1: The circuit design of car controlling system with Arduino using hand gestures recognition. Figure S2: The circuit design of mobile application to control car.

Author Contributions: All of the authors contributed in the literature search, figures, study design, data collection, data analysis, data interpretation, and writing etc. Such as, S.U., Z.M., M.A. and H.A.M. designed the devices and carried out the experimental work. S.U., Z.M., S.L., A.M. and H.A.M. analyzed the data and interpreted the results. H.A.M. and Z.M. drafted the manuscript with the input from the others. H.A.M. supervised the project.

Funding: This work is supported by Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Pakistan. H.A.M. acknowledges the support of the Postdoctoral Science Foundation of China at Southeast University, Nanjing, China, under Postdoctoral number 201557.

Acknowledgments: The authors would like to thank the Editor and the anonymous reviewers for their insightful comments and constructive suggestions that certainly improved the quality of this paper. The authors thank Majid Saleem, Zeeshan Ilyas, Abdullah Zulfiqar from Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan and Zaid Mumtaz from Department of Computer Science, Superior University, Lahore, 55150, Pakistan for critical discussions.

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

References

1. Qureshi, M.O.; Syed, R.S. The impact of robotics on employment and motivation of employees in the service sector, with special reference to health care. *Saf. Health Work* **2014**, *5*, 198–202. [CrossRef] [PubMed]
2. Fernando, Y.A.; Mathath, A.; Murshid, M.A. Improving productivity: A review of robotic applications in food industry. *Int. J. Rob. Appl. Technol.* **2016**, *4*, 43–62. [CrossRef]
3. Siramshetti, B.K.; Priyanka, K.; Rajitha, C. I won't fall down; Edge detector robot. *Int. J. Sci. Eng. Technol. Res.* **2017**, *6*, 113–116.
4. Pakdaman, M.; Sanaatiyan, M.M.; Ghahroudi, M.R. A line follower robot from design to implementation: Technical issues and problems. In Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, 26–28 February 2010; pp. 5–9.
5. Singh, R.R.; Das, R.; Pal, K. A novel modus of hand gesture controlled wireless robot. *Int. J. Res. Appl. Sci. Eng. Technol.* **2017**, *5*, 641–647.
6. Fang, B.; Sun, F.; Liu, H.; Liu, C. 3D human gesture capturing and recognition by the IMMU-based data glove. *Neurocomputing* **2018**, *277*, 198–207. [CrossRef]

7. Shi, Y.; Taib, R.; Lichman, S. Gesturecam: A smart camera for gesture recognition and gesture-controlled web navigation. In Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 5–8 December 2006; pp. 1–6.
8. Liu, T.; Luo, X.; Liu, J.; Cui, H. Compressive infrared sensing for arm gesture acquisition and recognition. In Proceedings of the IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015; pp. 1882–1886.
9. Dahl, T.S.; Gibbons, P.; Jones, O. Identification and production of simple tactile gestures. In Proceedings of the IEEE International Conference on Human-Robot Interaction Workshop on Advances in Tactile Sensing and Touch based Human-Robot Interaction, Boston, MA, USA, 5–8 March 2012. [CrossRef]
10. Cao, S.; Yang, P.; Li, X.; Chen, M.; Zhu, P. iPand: Accurate gesture input with smart acoustic sensing on hand. In Proceedings of the 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Hong Kong, China, 11–13 June 2018; pp. 1–3. [CrossRef]
11. Jost, C.; De Loor, P.; Nédélec, L.; Bevacqua, E.; Stanković, I. Real-time gesture recognition based on motion quality analysis. In Proceedings of the 7th International Conferences on Intelligent Technologies for Interactive Entertainment (INTETAIN), Turin, Italy, 10–12 June 2015; pp. 47–56.
12. Ju, M.H.; Kang, H.B. Emotional interaction with a robot using facial expressions, face pose and hand gestures. *Int. J. Adv. Rob. Syst.* **2012**, *9*, 1–13. [CrossRef]
13. Chen, C.-H.; Lee, I.-J.; Lin, L.-Y. Augmented reality-based self-facial modeling to promote the emotional expression and social skills of adolescents with autism spectrum disorders. *Res. Dev. Disabil.* **2015**, *36*, 396–403. [CrossRef] [PubMed]
14. Vyas, K.K.; Pareek, A.; Vyas, S. Gesture recognition and control. *Int. J. Recent Innov. Trends Comput. Commun.* **2013**, *1*, 575–581.
15. Cervantes-Villanueva, J.; Carrillo-Zapata, D.; Terroso-Saenz, F.; Valdes-Vela, M.; Skarmeta, A.F. Vehicle maneuver detection with accelerometer-based classification. *Sensors* **2016**, *16*, 1618. [CrossRef]
16. Aggarwal, L.; Gaur, V.; Verma, P. Design and implementation of a wireless gesture controlled robotic arm with vision. *Int. J. Comput. Appl.* **2013**, *79*, 39–43. [CrossRef]
17. Patel, N.K.; Patel, S.B.; Ammar, M.M. Accelerometer based gesture controlled wheel chair with GPS, GSM navigation. *Int. J. Innov. Emerg. Res. Eng.* **2015**, *2*, 110–113.
18. Goyal, D.; Saini, S.P.S. Accelerometer based hand gesture controlled wheelchair. *Int. J. Emerg. Technol.* **2013**, *4*, 15–20.
19. Jena, S.P.; Nayak, S.K.; Sahoo, S.K.; Sahoo, S.R.; Dash, S.; Sahoo, S.K. Accelerometer based gesture controlled robot using arduino. *Int. J. Eng. Sci. Res. Technol.* **2015**, *4*, 469–475.
20. Budheliya, C.S.; Solanki, R.K.; Acharya, H.D.; Thanki, P.P.; Ravia, J.K. Accelerometer based gesture controlled robot with robotic arm. *Int. J. Innov. Res. Sci. Technol.* **2017**, *3*, 92–97.
21. Mojeebi, T.; Tulo, S.K. Accelerometer gesture controlled robot using Arduino. *Int. J. Eng. Technol.* **2016**, *3*, 38–41.
22. Jayabala, P. Design and implementation of gesture controlled robotic arm for industrial applications. *Int. J. Sci. Res.* **2018**, *3*, 202–209.
23. Pławiak, P.; Sośnicki, T.; Niedźwiecki, M.; Tabor, Z.; Rzecki, K. Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms. *IEEE Trans. Ind. Inf.* **2016**, *12*, 1104–1113. [CrossRef]
24. Rzecki, K.; Paweł, P.; Niedźwiecki, M.; Sośnicki, T.; Leśkow, J.; Ciesielski, M. Person recognition based on touch screen gestures using computational intelligence methods. *Inf. Sci.* **2017**, *415*, 70–84. [CrossRef]
25. Jawalekar, P.A. Robot control by using human hand gestures. *Int. Res. J. Eng. Technol.* **2018**, *5*, 389–391.
26. Patil, P.V.; Shete, M.B.; Padalkar, T.M. Wireless hand gesture robot using accelerometer. *Int. Res. J. Eng. Technol.* **2016**, *3*, 353–356.
27. Verma, S. Hand gestures remote controlled robotic arm. *Adv. Electron. Electr. Eng.* **2013**, *3*, 601–606.
28. Suriya, T.S.U.; Khan, S.; Selvaganapathy, S.; Solaimani; Prasannakumar, V. Gesture controlled prosthetic arm. *Int. J. Adv. Res. Dev.* **2017**, *2*, 56–59.
29. Jhaveri, R.B.; Mehta, H.M.; Gala, P.T. Gesture controlled robot. *Int. J. Electron. Electr. Compute. Syst.* **2015**, *4*, 15–19.
30. Rao, S.; Rajasekhar, C. Password based hand gesture controlled robot. *Int. J. Eng. Res. Appl.* **2016**, *6*, 63–69.

31. Setia, A.; Mittal, S.; Nigam, P.; Singh, S.; Ganwar, S. Hand gesture recognition based robot using accelerometer sensor. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* **2015**, *4*, 4470–4476.
32. Swetha, N. Design and implementation of accelerometer based robot motion and speed control with obstacle detection. *Int. J. Sci. Eng. Technol.* **2013**, *2*, 749–755.
33. Mishra, A.; Malhotra, S.; Pradesh, U. Design of hand glove for wireless gesture control of robot. *Int. J. Pure Appl. Math.* **2017**, *114*, 69–79.
34. Verma, S. Android app controlled bluetooth robot. *Int. J. Comput. Appl.* **2016**, *152*, 35–40. [CrossRef]
35. Maity, A.; Paul, A.; Goswami, P.; Bhattacharya, A. Android application based bluetooth controlled robotic car. *Int. J. Intell. Inf. Syst.* **2017**, *6*, 62–66. [CrossRef]
36. Pahuja, R.; Kumar, N. Android mobile phone controlled bluetooth robot using 8051 microcontroller. *Int. J. Sci. Eng. Res.* **2014**, *2*, 14–17.
37. Kumar, R.; Ushapreethi, P.; Kubade, P.R.; Kulkarni, H.B. Android phone controlled bluetooth robot. *Int. Res. J. Eng. Technol.* **2016**, *3*, 104–114.
38. Roy, S.; Wangchuk, T.P.; Bhatt, R. Arduino based bluetooth controlled robot. *Int. J. Eng. Trends Technol.* **2016**, *32*, 216–219.
39. Kanere, S.; Shinde, R.; Tornekar, V.; Zagade, D.; Sonar, V.S. Bluetooth controlled car system. *Int. J. Adv. Res. Comput. Commun. Eng.* **2017**, *6*, 44–46. [CrossRef]
40. Gandotra, S.; Sharma, B.; Mahajan, S.; Motup, T.; Choudhary, T.; Thakur, P. Bluetooth controlled RC car using arduino. *Int. J. Interdiscip. Res.* **2016**, *2*, 144–147.
41. Rai, N.; Rasaily, D.; Wangchuk, T.R.; Gurung, M.; Khawas, R.K. Bluetooth remote controlled car using Arduino. *Int. J. Eng. Trends Technol.* **2016**, *33*, 381–384. [CrossRef]
42. Kannan, K.; Selvakumar, J. Arduino based voice controlled robot. *Int. Res. J. Eng. Technol.* **2015**, *2*, 49–55.
43. Rashid, H.; Ahmed, I.U.; Osman, S.B.; Newaz, B.; Rasheduzzaman, M.; Reza, S.M.T. Design and implementation of a voice controlled robot with human interaction ability. In Proceedings of the International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering, Rajshahi, Bangladesh, 26–27 January 2017; Volume 65, pp. 148–151.
44. Memon, Y.A.; Motan, I.; Akbar, M.A.; Hameed, S.; Hasan, M.U. Speech recognition system for a voice controlled robot with real time obstacle detection and avoidance. *Int. J. Electr. Electron. Data Commun.* **2016**, *4*, 33–37.
45. Zope, S.; Muluk, P.; Mohite, R.; Lanke, A.; Bamankar, M. Voice control robot using android application. *Int. J. Interdiscip. Res.* **2017**, *3*, 1723–1726.
46. Harshad, P.; Amol, J.; Dnyaneshwar, K.; Bhosale, A.C. Voice control robot using Arduino pick and place object. *Int. J. Innov. Eng. Res. Technol.* **2017**, *4*, 27–30.
47. Kumar, A.; Chauhan, R. Voice controlled robot. *Int. J. Innov. Res. Technol.* **2014**, *1*, 338–344.
48. Saravanan, D.; Parthiban, R.; Archanaa, G.I. Voice controlled robotic car using Arduino for smart agriculture. *Int. J. Pure Math.* **2018**, *118*, 2097–2105.
49. Chikhale, V.; Gharat, R.; Gogate, S.; Amireddy, R. Voice controlled robotic system using Arduino microcontroller. *Int. J. New Technol. Res.* **2017**, *3*, 92–94.
50. Louis, L. Working principle of Arduino and using it as a tool for study and research. *Int. J. Control Autom. Commun. Syst.* **2016**, *1*, 21–29.
51. Punetha, D.; Kumar, N.; Mehta, V. Development and applications of line following robot based health care management system. *Int. J. Adv. Res. Comput. Technol.* **2013**, *2*, 2446–2450.
52. Ebiesuwa, O.O.; Adekunle, Y.A.; Akinyemi, L.A.; Oyerinde, O.D. Line follower robot using a sophisticated sensor approach. *Int. J. Eng. Res. Technol.* **2013**, *2*, 1980–1982.
53. Gumus, O.; Topaloglu, M.; Ozcelik, D. The use of computer controlled line follower robots in public transport. *Procedia Comput. Sci.* **2016**, *102*, 202–208. [CrossRef]
54. Mi, J.; Takahashi, Y. An design of hf-band rfid system with multiple readers and passive tags for indoor mobile robot self-localization. *Sensors* **2016**, *16*, 1200. [CrossRef] [PubMed]
55. Chavan, A.S.; Pansare, D.D.; Kadam, S.P.; Mayekar, N.K.; Jha, K.V.; Bhagwat, P.R. Design of accident prevention system using QRD 1114 and CNY 70 Sensors. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 525–530.
56. Mohamed, A.; Yang, C.; Cangelosi, A. Stereo vision based object tracking control for a movable robot head. *Int. Fed. Autom. Control* **2016**, *49*, 155–162. [CrossRef]
57. Pati, C.S.; Kala, R. Vision-based robot following using pid control. *Technologies* **2017**, *5*, 34. [CrossRef]

58. Nadiger, N.; Madhusudan, K.; Naik, S. humanoids implementation using sixth sense. *Int. J. Inf. Syst. Comput. Sci.* **2012**, *1*, 31–34.
59. Purohit, R.; Vyas, S.; Mathur, P. AI and its application: Sixth sense technology. *Int. J. Emerg. Trends Technol. Comput. Sci.* **2013**, *2*, 184–186.
60. Daroga, R.; Pandey, N. Sixth Sense Technology & Its Applications. *Int. J. Sci. Res. Publ.* **2015**, *5*, 1–4.
61. Dhanalakshmi, V.; Jayasri, D.; Vishnupriya, R. Zigbee based wireless intelligent security robot for defence camp. *Int. J. Adv. Eng. Glob. Technol.* **2014**, *2*, 576–583.
62. Patel, A.; Chaudhari, K.; Patel, D. Touch screen controlled multipurpose spy robot using zigbee. *Int. J. Adv. Res. Comput. Eng. Technol.* **2014**, *3*, 1058–1062.
63. Premkumar, M. Unmanned multi-functional robot using zigbee adopter network for defense application. *Int. J. Adv. Res. Compute. Eng. Technol.* **2013**, *2*, 47–55.
64. Song, B.; Lu, X.; Bai, X. Zigbee based wireless sensor and actuator network for service robot intelligent space. *Wirel. Sens. Netw.* **2012**, *4*, 235–242. [CrossRef]
65. Iswarya, P.; Ramarao, D.; Kumar, B.D.; Kumar, K.D.; Manikyalarao, T. Obstacle avoidance robot using Arduino. *Int. J. Adv. Res. Sci. Eng.* **2017**, *6*, 634–638.
66. Bhagat, K.; Deshmukh, S.; Dhonde, S.; Ghag, S. Obstacle Avoidance Robot. *Int. J. Sci. Eng. Technol. Res.* **2016**, *5*, 439–442.
67. Ankit, V.; Jigar, P.; Savan, V. Obstacle avoidance robotic vehicle using ultrasonic sensor, android and bluetooth for obstacle detection. *Int. Res. J. Eng. Technol.* **2016**, *3*, 339–348.
68. Dumbre, K.; Ganeshkar, S.; Dhekne, A. Robotic vehicle control using internet via webpage and keyboard. *Int. J. Comput. Appl.* **2015**, *114*, 15–19. [CrossRef]
69. Kadir, W.M.H.W.; Samin, R.E.; Ibrahim, B.S.K. Internet controlled robotic arm. *Int. Symp. Rob. Intell. Sens.* **2012**, *41*, 1065–1071. [CrossRef]
70. Patil, A.D.; Kadiri, H.I.; Joshi, A.S.; Wani, A.B. IoT based remote access human control robot using mems sensor. *Int. J. Comput. Sci. Mob. Comput.* **2016**, *5*, 816–826.
71. Kalaiarasi, D.; Pavithra, S.; Pratheeba, S.; Priyaadharshini, R.L. IoT based motion control system of a robotic car. *Int. Res. J. Eng. Technol.* **2018**, *5*, 3073–3076.
72. Nanda Kishor, M.; Prasad, A.P.; Vinod Kumar, M.; Manikanta, R.; Geetha, M.N.; Kavyashree, B. IoT based smart car parking using line following robot. *Int. J. Electr. Electron. Data. Commun.* **2017**, *5*, 46–48.
73. Nayyar, A.; Puri, V.; Nguyen, N.G.; Le, D.N. Smart surveillance robot for real-time monitoring and control system in environment and industrial applications. *Inf. Syst. Des. Intell. Appl. Adv. Intell. Syst. Comput.* **2018**, 229–243. [CrossRef]
74. Ananthapadmanabhan, J.; Abraham, A.M.; George, L.M.; Saji, V.A.; Anil, A.R. Smart robotic assistant using IoT. *Int. Res. J. Eng. Technol.* **2017**, *4*, 857–859.
75. Waldherr, S.; Romero, R.; Thrun, S. A gesture based interface for human-robot interaction. *Auton. Rob. Springer* **2000**, *9*, 151–173. [CrossRef]
76. Perrin, S.; Cassinelli, A.; Ishikawa, M. Gesture recognition using laser-based tracing system. In Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 19 May 2004; pp. 541–546.
77. Google Play. Arduino Bluetooth Control. Available online: https://play.google.com/store/apps/details?id=com.broxcode.arduinobluetoothfree&hl=en (accessed on 31 December 2018).
78. Arduino Uno Rev3. Available online: https://store.arduino.cc/usa/arduino-uno-rev3 (accessed on 31 December 2018).
79. Compenets101. Arduino Nano. Available online: https://components101.com/microcontrollers/arduino-nano (accessed on 31 December 2018).
80. SunRom. Gyro + Accelerometer Sensor, 3 Axis based on MPU-6050. Available online: https://www.sunrom.com/p/gyro-accelerometer-sensor-3-axis-based-on-mpu-6050 (accessed on 31 December 2018).
81. ElProCus. RF Module—Transmitter & Receiver. Available online: https://www.elprocus.com/rf-module-transmitter-receiver/ (accessed on 31 December 2018).
82. SunFounder. L293D Motor Driver Shield. Available online: http://wiki.sunfounder.cc/index.php?title=L293D_Motor_Driver_Shield (accessed on 31 December 2018).
83. Itead. Serial Port Bluetooth Module HC-05. Available online: https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05 (accessed on 31 December 2018).

84. How to Mechatronics. L298 Motor Module. Available online: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/ (accessed on 31 December 2018).
85. Youtube. An Automated Robot Car Control System with Hand Gestures and Mobile Application Using Arduino. Available online: https://www.youtube.com/watch?v=3Nqz9jewack&feature=youtu.be (accessed on 31 December 2018).
86. Reddy, B.R.; Mahender, E. Speech to text conversion using android platform. *Int. J. Eng. Res. Appl.* **2013**, *3*, 253–258.
87. Insteon. Insteon WiFi Cameras. Available online: https://www.insteon.com/wifi-cameras/ (accessed on 31 December 2018).
88. Krotak, T.; Simlova, M. The analysis of the acceleration of the vehicle for assessing the condition of the driver. In Proceedings of the IEEE Intelligent Vehicles Symposium, Alcala de Henares, Spain, 3–7 June 2012; pp. 571–576.
89. González, A.; Olazagoitia, J.L.; Vinolas, J. A low-cost data acquisition system for automobile dynamics applications. *Sensors* **2018**, *18*, 366. [CrossRef]