Research Article

Smart Drinking Control System Based on Raspberry Pi

Sung-Bae Kim[1], Ki-Young Lee[2], Sung-Ho Hwang[3], Yeon-Man Jeong[4], Jeong-Jin Kang[5]

[1,2] Professor, Department of Medical IT, Eulji University, Seongnam, Korea
[3] Professor, Division of Electronics, Information and Communication Engineering, Kangwon National University, Samcheok, Korea
[4] Professor, Department of Information and Communication Engineering, Gangneung-Wonju National University, Wonju, Korea
[5] Professor, Department of Information and Communication, Dong Seoul University, Seongnam, Korea

Corresponding Author:: Ki-Young Lee
Professor, Department of Medical IT, Eulji University, Seongnam, Korea, kylee@eulji.ac.kr

Abstract: Currently, police officers randomly arrange sobriety checkpoints and verify drunk drivers on the road through breath test. Regardless of such efforts of police officers, people share and avoid the locations of driving under the influence (DUI) checkpoint cunningly. The purpose of this study is to design and implement a system that prevents drunk drivers from taking their car out and easily connects the drivers to a chauffeur service provider using Arduino, Voyager face recognition camera, and OpenCV. This system enables an efficient allocation of police officers who check drunk driving based on DUI parking lot checkpoints and creates profit by connecting drivers to a chauffeur service provider, resulting in the reduction of drunk driving victims.

Keywords: OpenCV, Raspberry Pi, Arduino, Voyager, Tesseract

1. Introduction

The number of vehicles travelling on roads is increasing rapidly every year, and the structural improvement of roads and increase in the number of facilities enhance the road condition of Korea to the level of advanced countries. However, the safety awareness of drivers still remains at the same level as that of 20 years ago. The number of traffic accidents caused by drunk driving is increasing every year. Such accidents create severe mental, physical, and property damages to people. Since the magnitude of such drunk driving is very high, the government and the police are making efforts to prevent drunk driving in various directions. However, regardless of such efforts, accidents caused by drunk driving are not decreasing in number drastically **(Won Joong Kim, 2009).** As a result, the penalties for drunk driving are being reinforced continuously as a policy to control drunk driving **(Jin Hyung Kim, 2011)**. However, the hidden propensity of drunk driving is very high, so most drunk driving cases are not revealed properly on the official statistical figures. It has been reported that the probability of detecting drunk drivers is merely 0.0005% **(Chan Geol Park, 2011).**

In this study, a system that prevents drunk drivers from taking their car out in the parking lot and connects the drivers to a chauffeur service provider using Arduino and Raspberry Pi is suggested as a measure to eliminate drunk driving for reducing the incidence rate of accidents caused by drunk driving.

2. Related Works

2.1. OpenCV

Open Source Computer Vision (OpenCV) is a programming library aiming at real-time computer vision. It was originally developed by Intel. This library focuses on real-time image processing **(Dong-Hwan Gong, et. al, 2018; Sang-hee Yun, et. al, 2017).**

2.2. Raspberry Pi

This is a credit card-sized microscopic and super cheap PC developed by the Raspberry Pi Foundation in U.K. to provide basic computer and science education in schools. Unlike Arduino, Raspberry Pi can be used as a PC by simply connecting a keyboard, a mouse, and a monitor. This product emphasizes the fact that it is similar to normal desktop PCs **(Jeong-Hoon Lee, et. al, 2019; Taejoon Park & Jaesang Cha, 2018).**

2.3. Arduino

Arduino refers to open source-based boards (products) completed by a single board microcontroller and the related development tools and environment. It provides Integrated Development Environment (IDE) for Arduino as well as software development and executable code uploads **(Young-ho Ko, Gyu-Seong Heo & Sang-Hyun Lee, 2019; Jong-Youel Park & Young-Hyun Chang, 2018).**

2.4. Voyager

Voyager is a universal asynchronous receiver-transmitter (UART) communication-based face recognition module that can provide an optimal solution for the implementation of face recognition system using Raspberry Pi, Arduino, and Odroid and reduce the development time of IT security system **(Seongwon Min, Jong-Yong Lee & Kye-Dong Jung, 2017; Rethina kumar, Gopinath Ganapathy & Jeong-Jin Kang, 2018).**

2.5. Tesseract

Tesseract is an optical character recognition engine for various operating systems. This software is a free software distributed in accordance with Apache License Version 2.0, and its development was sponsored by Google since 2006 **(Kack-He Lee, et. al, 2017; Yun-Taek Lim, Da-Hye Kim & Min-Jeong Koo, 2016).**

3. Smart Drinking Control System

3.1. System Design

In this study, license plate recognition function and face recognition function applied to existing parking lots are necessary in order to implement the parking lot breath test gate system. The license plate recognition process is divided into three steps including license plate detection, individual character extraction, and character recognition. To detect the license plate area from a vehicle image effectively, it is important to obtain a vehicle image of which relatively constant brightness level is maintained and an undistorted license plate image **(Jin Ho Kim, 2011).** The relevant system design drawing is shown in Figure 1.

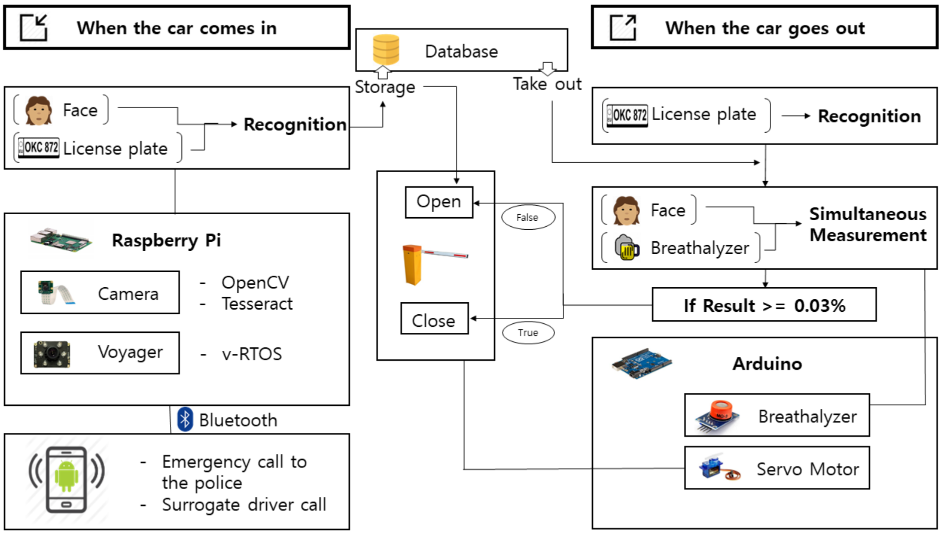


Figure 1. System Architecture

In the design drawing shown in Figure 1 above, a raspberry camera is used to capture the picture of an entering vehicle. License plate is detected from the captured picture using OpenCV and individual characters in the license plate are extracted. The extracted characters are recognized through Tesseract. In addition, face information is recognized through a Voyager camera and both the recognized license plate information and face information are stored in database. When the vehicle exits, the license plate is recognized and drawn from the database. The driver's face is checked whether it matches the saved face information, and the level of alcohol in the blood is measured through Arduino alcohol measurement sensor at the same time. If the level of alcohol in the blood is equal to or less than 0.03%, the parking lot gate will open via Arduino servomotor. If the level is greater than 0.03%, the gate will not open and the driver will be asked whether he will request for a chauffeur service via Arduino LCD module and Arduino switch module. Then, the system connects the driver to a chauffeur service provider via Bluetooth module. This system is intended for reducing the accident rates caused by drunk driving. The flow of the relevant functions is as shown in Figure 2.

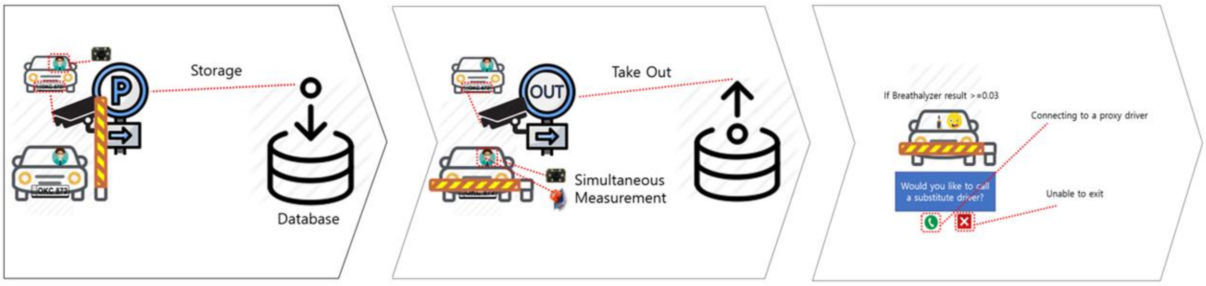
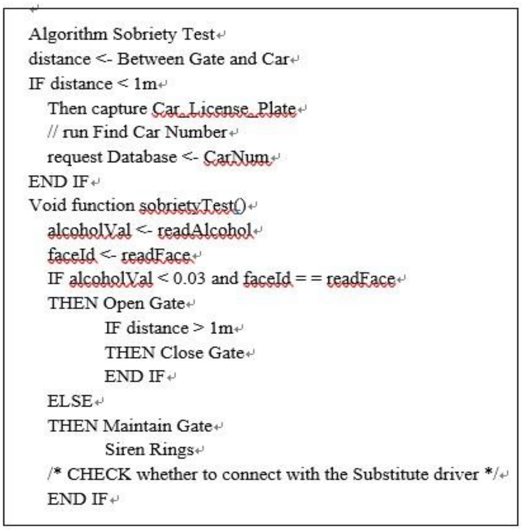

Storage
Database
Take Out
Simultaneous Measurement
If Breathalyzer result >=0.03
Connecting to a proxy driver
Would you like to call a substitute driver?
Unable to exit

Figure 2. System Flow

3.2. System Implementation

The system in this study was implemented on Window 10 64bit operating system, and Jupyter Notebook, Arduino modules, and Raspberry Pi modules were used to implement such system. When the license plate and face information are collected, these information are saved in database. Face recognition and measuring the level of alcohol in the blood should be carried out in order to check the alcohol consumption status when a vehicle exits. The algorithm for implementing such process is as shown in Figure 3. An ultrasonic sensor is used to check whether a vehicle approaches or not. It is checked whether the identified license plate matches the face information in the same column. This is a step where the gate will open if the level of alcohol in the blood is equal to or less than 0.03%, and if it is higher than 0.03%, a warning alarm will sound, checking whether the driver will call a chauffeur service.



```
Algorithm Sobriety Test
distance <- Between Gate and Car
IF distance < 1m
   Then capture Car_License_Plate
   // run Find Car Number
   request Database <- CarNum
END IF
Void function sobrietyTest()
   alcoholVal <- readAlcohol
   faceId <- readFace
   IF alcoholVal < 0.03 and faceId = = readFace
   THEN Open Gate
            IF distance > 1m
            THEN Close Gate
            END IF
   ELSE
   THEN Maintain Gate
            Siren Rings
   /* CHECK whether to connect with the Substitute driver */
   END IF
```

Figure 3. Sobriety Test Algorithm

3.3. System Implementation Results

The system implementation result suggested in this study is as shown in Figures 4, 5, 6, and 7. Figure 4 shows the camera and Raspberry Pi connection for capturing a license plate. In Figure 5, the square surrounding the contour lines is obtained, and the x and y coordinates, width, and height of the square are stored in order to find the location of the license plate after the vehicle is captured. Next, the minimum and maximum values of area, width, height, and horizontal to vertical ratio of the bounding square are determined and values that are likely to represent the license plate are stored in array.

Figure 6 shows the image of the final result after finding the contour lines that are likely to represent the license plate. This is the picture of the license plate identified from the whole image. Figure 7 shows the composition of Arduino ultrasonic sensor, alcohol sensor, and servomotor. The ultrasonic sensor measures the distance and if it is within a certain distance, the alcohol sensor detects alcohol and the servomotor plays a role of a gate.

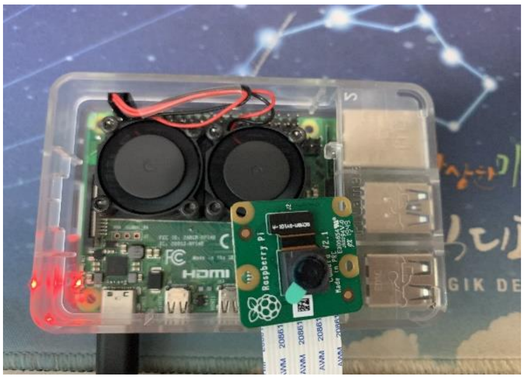


Figure 4. Raspberry Pi Module

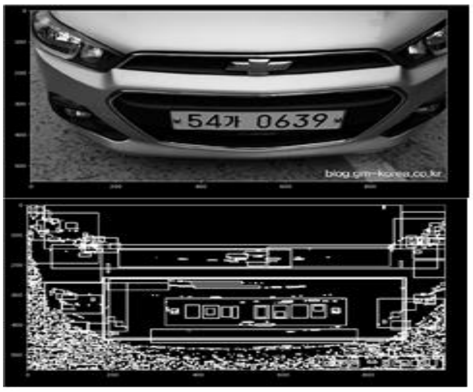
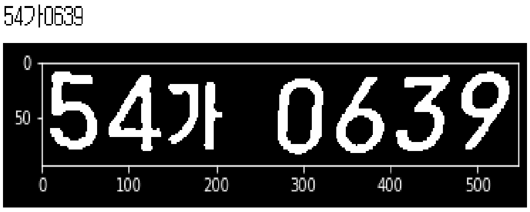


Figure 5. Surrounded Rectangle



```
54가0639
```

Figure 6. Found License Plate

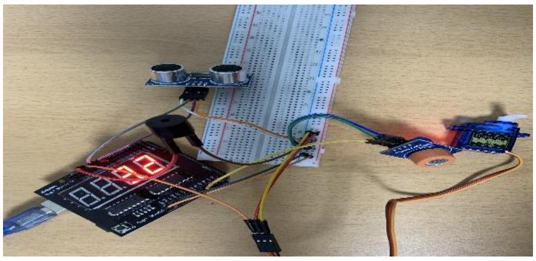
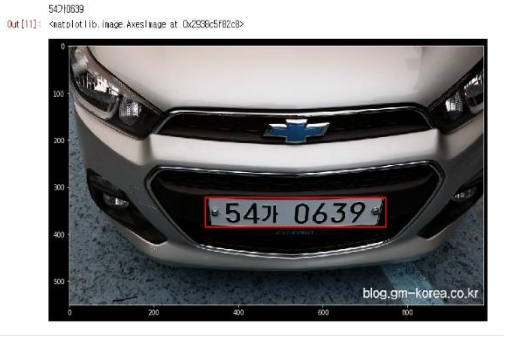


Figure 7. Arduino Module

4. Performance Evaluation

The performance evaluation for drinking status through license plate recognition and Arduino alcohol sensor is as shown in Figures 8, 9, 10, and 11.



```
54가0639
Out[11]: <matplotlib.image.AxesImage at 0x2938c5f82c8>
```

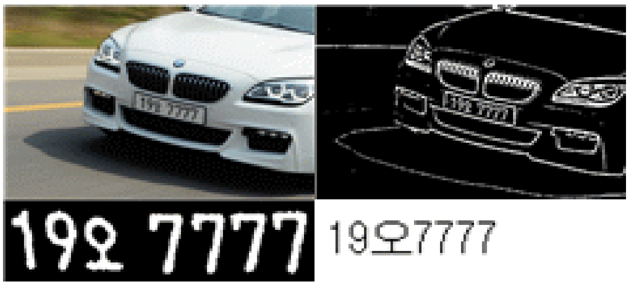
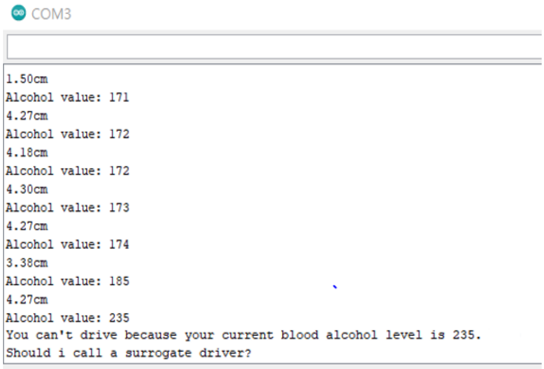


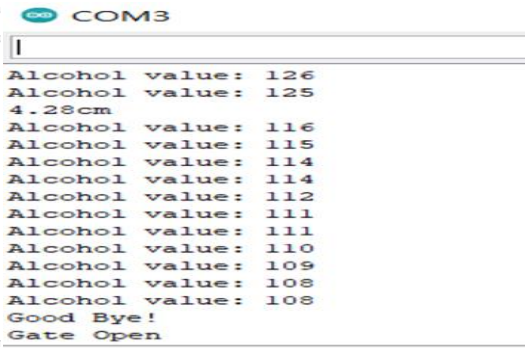
Figure 8-1. License Plate to Character1 **Figure 8-2.** License Plate to Character2

As shown in Figure 8-1, the license plate has been identified from the picture and it has been successfully converted and displayed into text. Figure 8-2 shows the test carried out using a different vehicle image. It is confirmed that the license plate was also identified successfully from such image and the result was displayed. Figure 9 shows the case of a drunk driver with the measured level of alcohol in the blood exceeds 200, so the system does not open the gate and asks the driver to call a chauffeur service provider or not. In Figure 10, the measured level of alcohol in the blood was normal, so the gate opened via servomotor. To test the performance of Arduino alcohol sensor, the result measured using a commercially available breathalyzer after drinking was compared and analyzed for four days.

The comparison graphs for alcohol consumption are as shown in Figure 11-1. The red line indicates sobriety test criteria, the blue line indicates breathalyzer, and the orange line indicates Arduino alcohol sensor. There is a thin margin of error. However, there was no problem for determining the alcohol consumption status. Figure 11-2 shows the graph at the time of no alcohol consumption. There is also a thin margin of error, but the result measured by Arduino alcohol sensor is relatively higher.



```
1.50cm
Alcohol value: 171
4.27cm
Alcohol value: 172
4.18cm
Alcohol value: 172
4.30cm
Alcohol value: 173
4.27cm
Alcohol value: 174
3.38cm
Alcohol value: 185
4.27cm
Alcohol value: 235
You can't drive because your current blood alcohol level is 235.
Should i call a surrogate driver?
```



```
Alcohol value: 126
Alcohol value: 125
4.28cm
Alcohol value: 116
Alcohol value: 115
Alcohol value: 114
Alcohol value: 114
Alcohol value: 112
Alcohol value: 111
Alcohol value: 111
Alcohol value: 110
Alcohol value: 109
Alcohol value: 108
Alcohol value: 108
Good Bye!
Gate Open
```

Figure 9. After Drinking Sobriety Test

Figure 10. Gate Open

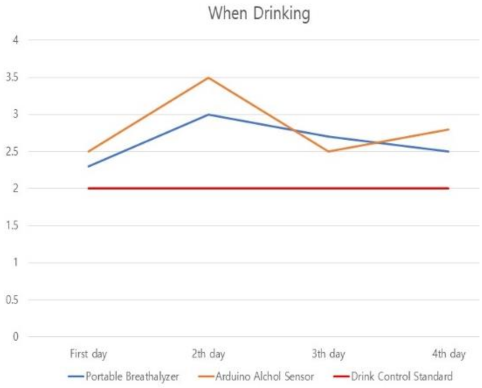


Figure 11-1. After Drinking Comparison Graph

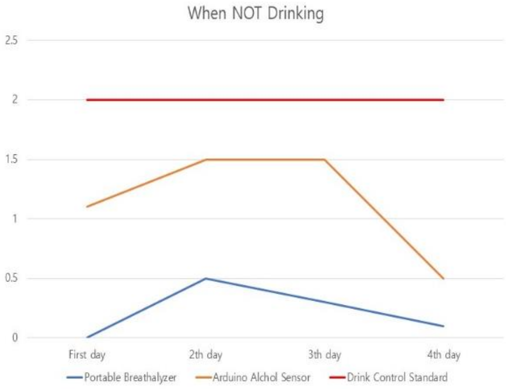


Figure 11-2. Not Drinking Comparison Graph

5. Conclusion

In this study, a vehicle picture was captured and imported to recognize the license plate number through Raspberry Pi and OpenCV, and the driver's face information was verified through Voyager face recognition camera. In addition, status of an approaching vehicle, alcohol detection, and gate functions were implemented using Arduino. Even when the sobriety test is not carried out by the police, this system prevents the drunk driver's vehicle from exiting the parking lot and allows drivers to call chauffeur service easily, providing an effect to reduce the rate of drunk driving.

It was intended to check the QR code for the driver license in order to prevent an unlicensed chauffeur service provider from checking the level of alcohol in the blood on behalf of the driver. However, it was difficult to implement such function due to the limitation that the QR code for the driver's license was not generalized. A study on the method for preventing an unlicensed chauffeur service provider will be carried out in future.

References

A. Chan Geol Park. (2011). Reasonable Countermeasure on Sobriety Test Disobedience Offense Application. *Korean Criminological Review*, 22(3), 139-173.

B. Dong-Hwan Gong, Seung-Jung Shin. (2018). Comparative Analysis between Super Loop and FreeRTOS Methods for Arduino Multitasking. *The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC)*, 18(6), 133.

C. Jeong-Hoon Lee, Seung-Hun Jeong, Young-Gon Kim. (2019). Mobile-IoT System for Payment Efficiency and Convenience of Offline Shopping. *The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC)*, 19(1), 289-294.

D. Jin Ho Kim. (2011). Distortion Invariant Vehicle License Plate Extraction and Recognition Algorithm. *The Journal of The Korea Content Association*, 11(3), 1-8.

E. Jin Hyung Kim. (2011). The Relationship between Drinking/Driving Behavior and Drunk Driving. *The Journal of The Korean Urban Management Association*, 24(4), 203-223.

F. Jong-Youel Park, Young-Hyun Chang. (2018). Study on Arduino Kit VR contents modularization based on virtualization technology in software education field. *The Journal of the Convergence on Culture Technology (JCCT)*, 4(3), 293-298.

G. Kack-He Lee, Min-Wo Lee, Kyung-Wan Koo, Jae-Sang Cha. (2017). Development of IoT-based data acquisiton device (Gateway) for context-awarenes of hospital facilties. *The Journal of the Convergence on Culture Technology (JCCT)*, 3(4), 181-184.

H. Rethina kumar, Gopinath Ganapathy, Jeong-Jin Kang. (2018). A Novel Architecture for Mobile Crowd and Cloud computing for Health care. *International Journal of Advanced Culture Technology(IJACT)*, 6(4), 226-232.

I. Sang-hee Yun, Jeong-joon Kim, Gwang-il Jeon. (2017). A Proposal for expansion of the low-floor bus based on analysis of living area for the handicapped mobility people in Seoul Using R. *The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC)*, 17(6), 195-201.

J. Seongwon Min, Jong-Yong Lee, Kye-Dong Jung. (2017). Design of Cordinator Based on Android for Data Colection in Body Sensor Network. *International Journal of Advanced Culture Technology(IJACT)*, 5(2), 98-105.

K. Taejoon Park, Jaesang Cha. (2018). Development of IoT based Real-Time Complex Sensor Board for Managing Air Quality in Buildings. *International Journal of Internet, Broadcasting and Communication(IJIBC)*, 10(4), 75-82.

L. Won Joong Kim. (2009). A Study on measures to solve the problems of General Curbing on a Drunken Driver of Police. *Legal discussion April*, 6-22.

M. Young-ho Ko, Gyu-Seong Heo, Sang-Hyun Lee. (2019). A Study on Distributed System Construction and Numerical Calculation Using Raspberry Pi. *International Journal of Advanced Smart Convergence(IJASC)*, 8(4), 194-199.

N. Yun-Taek Lim, Da-Hye Kim, Min-Jeong Koo. (2016). Study about Recommended Styling System of Android-based Shopping mall. *The Journal of the Convergence on Culture Technology (JCCT)*, 2(4), 61-64.