EE595 Final Project Team11

Yeongbin Hwang
KAIST

1 Introduction

With the gradual development of science and technology, we can easily come into contact with electronic devices such as IoT(Internet of Things). Various devices such as smart watches and voice assistants appeared, and more and more IoT devices appeared. In 2021, IoT equipment will be 35.82 billion devices, and it is predicted that it will develop to about 75.44 billion by 2025 [8]. Among them, smart watch occupies a large proportion in the IoT market. The smartwatch market size is currently $59 billion, an increase of 18.7% compared to last year [7]. Smartwatches provide functions such as exercise tracking and heart rate measurement, which are functions for exercise, and includes an oximeter and ECG sensor for health management. Recently, functions such as fall detection to detect accident situations have also appeared. As such, the number of users is increasing by providing additional convenient functions for users.

The various functions of this smart watch are focused on functions for the convenience of people. In the first quarter of 2021, the total number of crimes in Korea is about 310,000, of which there are approximately 6,000 violent crimes [11]. Moreover, the arrest rate is low, so even if a crime occurs, a large number of criminals are not caught. In this way, we can be easily exposed to danger, and there are actually a lot of crimes. However, compared to that, the service that can prepare us for dangerous situations is insufficient. In addition, there is only research using smartphones and research that detects accident situations [9]. So, we thought what if we add a service that can detect dangerous situations to a smart watch that can be easily accessed around us?

In this project, we tried to detect dangerous situations through smartwatch. We created a service that detects two potentially dangerous situations through the IMU(Inertial Measurement Unit) sensor present in all smartwatches and sends them to the phone to alert the user. To show the feasibility of the service, we tested it using Arduino Nano 33 BLE Sense [2], and through this, the accuracy of voice and gesture recognition was measured to be 92 or 93%, implying that it will perform well when applied to real-world scenarios.

2 Background

2.1 Kalman Filter

Kalman filter is a linear optimal status estimation method, which is known as one of the most famous Bayesian filter theories [10]. It is an algorithm that uses a series of data observed over time, which contains noise and other inaccuracies, to estimates unknown variables with more accuracy. Because it has merits of real-time, fast, efficient and strong anti-interference, the Kalman filter has been widely applied. Therefore by using a Kalman filter, noisy accelerometer, gyro, and magnetometer data can be combined to obtain an accurate representation of orientation and position.

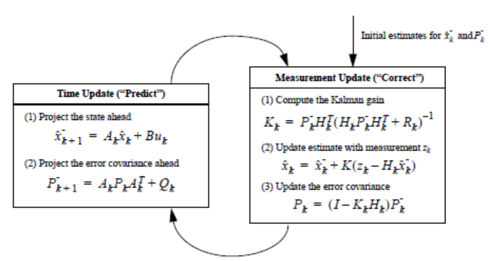

Figure 1: Operation of Kalman Filter.

Fig. 1 shows the operation of Kalman Filter [12]. The Kalman filter basically consists of two stages. In the first stage, a mathematical state model is used to make a prediction about the system state. In the next stage this state prediction is compared to measured state values. The difference between the predicted and measured state is moderated based on estimated noise and error in the system and measurements, and a state estimation is output. The output estimation is then used in conjunction with the mathematical state model to predict the future state during the next time update, and the cycle begins again.

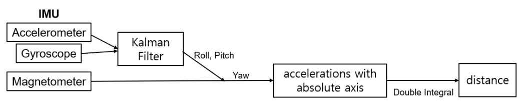

Figure 2: Algorithm for Trajectory Tracking.

2.2 TensorFlow Lite for Microcontrollers

TensorFlow Lite for microcontrollers is designed to run machine learning models on microcontrollers and other devices with only few kilobytes of memory [5]. The core runtime just fits in 16 KB on an Arm Cortex M3 and can run many basic models. It doesn't require operating system support, any standard C or C++ libraries, or dynamic memory allocation. The framework is available as an Arduino library. In addtion, it support Arduino Nano 33 BLE Sense. Therefore, we can adapt TensorFlow to Arduino by using TensorFlow Lite for Microcontrollers.

3 Implementation

3.1 Trajectory Tracking (Part 1)

In part 1, we implemented pointer trajectory tracking as precisely as possible. Specifically, when the initial position of the Arduino board is called the "reference point", we will measure the distance between the reference point and destination with linear movement and non-linear movement.

Fig. 2 shows the algorithm for Trajectory Tracking. First, we calculate the angle(roll, pitch, yaw) of Arduino position. To get roll, pitch angle, we calculate Arduino angle AngleX, AngleY with accelerometer data using the formula shown in Eq (1) and (2) [1], and also calculate the angle by integral the gyroscope data.

$$AngleX = \arctan\left(\frac{aX}{\sqrt{aY^2+aZ^2}}\right) \tag{1}$$

$$AngleY = \arctan\left(\frac{aY}{\sqrt{aX^2+aZ^2}}\right) \tag{2}$$

The angle obtained by the accelerometer has the advantage that the error does not increase over time and is limited to a certain range. Futhermore, the gyro detects changes in posture well, but the error accumulates and diverges over time. For that reason If we use the raw angle from the accelerometer and gyroscope, the result was too noisy and not accurate. Therefore we apply the Kalman Filter explained in the background section to improve the performance. Because the yaw angle of Arduino can't calculate by using accelerometer and gyroscope, we use magnetometer to calculate yaw value. Second, Remove roll, pitch and yaw angle to get accelerations with the absolute axis. Finally, We take double integral to compute location.

3.2 Mobile Bodyguard (Part 2)

Mobile Bodyguard is a service that detects a dangerous situation and sends a warning signal. The service was implemented using IMU sensor and microphone sensor. First, we define a dangerous situation and proceed using gesture and voice recognition to detect the situation. When a dangerous situation is detected, a warning screen is displayed on the UI by sending a danger signal to the notebook through BLE communication.

3.2.1 Define emergency situations.

As for the dangerous situation, a situation where only motion can be performed (situation 1) and a situation where voice and motion can be used (situation 2) were considered.
Situation 1 is a situation where people have been kidnapped or imprisoned and their hands and mouths are tied, where people can only move slightly.
Situation 2 is a situation in which a robber is attacked and people are threatened. Simple motions such as "help me" or raising a hand are possible.

3.2.2 Situation 1.

Gesture recognition. In this case, only a small amount of motion is possible, so we set the "shake" motion as a gesture to trigger a dangerous situation. In a situation where the hands are tied, other large motions are impossible, so it was decided to be a motion that can be easily executed. Three gestures were set by adding "rotate" and "hands-up" gestures as motions for comparison with this. A fully-connected neural network model was trained in Colab [6] by setting the angular velocities and accelerations in the x, y, and z axes that appear when these motions occur as features.

Challenge. If we train the model in this way and apply shake gesture recognition, we can see that the accuracy of gesture recognition is quite high. However, in the case of gesture recognition, there is a challenge that gesture can be easily recognized even in daily life and can be mistakenly recognized when not desired. Since an emergency alert should be sent only in a dangerous situation, practicality is reduced if detected in normal life.

Solutions. To resolve the challenge, we tried to reduce the misrecognition case by adding several methods.
First, detecting 5 shakes instead of once. Since one shake gesture recognition can be identified even in everyday situations, five shake gestures must occur before a dangerous situation is recognized. We set it to 5 because dangerous situations can still be acknowledged in everyday life if there are too few recognitions set, and it is difficult to trigger dangerous situations if there are too many. Second, set the time window to detect only continuous gestures. It is still not enough to consider only the recognition of five shake gestures. The reason is that if the shake gesture is recognized in multiple

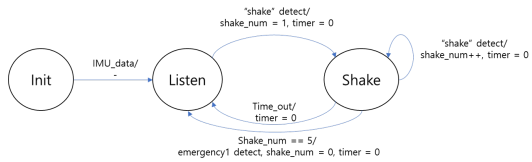

Figure 3: State machine of situation 1

situations once, and the total number of overlapping becomes 5, that situation can also be reached. So, we set the time window so that it is recognized as a dangerous situation only if shake is detected 5 times within that time. If two methods are considered and implemented, it can be expressed as a state machine like Fig. 3.

3.2.3 Situation 2.

Voice recognition. In this case, we set “help” as the voice to trigger the danger in a situation where the victim can speak a little. Even if the victim uses his/her voice, it is set to "help" because only a few words can be used to inform the dangerous situation without being detected. Three voices were set by adding "no" and "ok" as voices for comparison with this. The rms(root mean square) value was obtained by collecting 256 values from bytes that are generated when such a voice is recognized, and set as a feature, and trained as a fully-connected neural network model in colab.

Challenge. However, even with high accuracy in voice recognition, there is a challenge that the word help can be used in everyday life and can be mistakenly recognized as a dangerous situation.

Solutions. We added a few solutions to resolve the challenge to make it more robust. First, combines voice and gesture recognition. Gesture recognition has been added to prevent voice recognition from being recognized in normal, not dangerous situations. Therefore, after voice recognition, a specific gesture "hands-up" must also be detected to identify a dangerous situation and send a notification. In this case, it will be more robust to misrecognition because it uses two recognition techniques instead of one and has to pass two recognitions. Second, set the time window to detect only continuous voices and gestures. In this case as well, if the help voice is recognized and there is no time limit, it will be identified as a dangerous situation if the gesture occurs by mistake even once after the help is recognized. Therefore, it is recognized as a harzard only when the gesture occurs within a set time after the help voice is recognized. If we show this implementation as a state machine, Fig. 4 can be expressed together.

3.2.4 BLE communication.

BLE is intended to provide considerably reduced power consumption and cost while maintaining a similar communica-

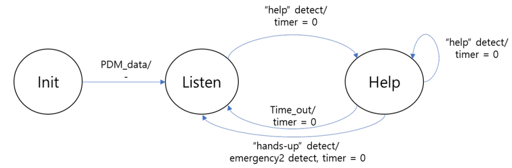

Figure 4: State machine of situation 2

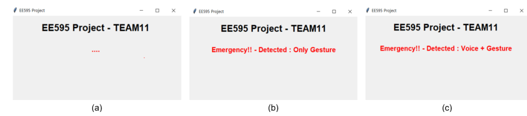

Figure 5: Emergency Alert UI using BLE communication (a) Nomally, (b) detected emergency situation1, (c) detected emergency situation2

tion range. When the board detects the emergency situation, it sends an emergency alert using BLE communication. We use a laptop as clients and it can receive data from peripheral devices. The Arduino board of our system is a server, so they transmit the emergency data to the client. On the laptop, we use Bleak which is a GATT client software [3]. It is designed to provide an asynchronous, cross-platform Python API to connect and communicate with other devices. By using this API, we can receive emergency data from our device and can display the emergency situation on the UI.

Fig. 5 shows the UI we implemented for emergency alerts. The board distinguishes whether it is situation 1 or 2, and sends an emergency alert, and the UI shows a dangerous situation on the screen according to it. Fig. 5 (b) is a UI screen when emergency situation 1 occurs, and Fig. 5 (c) is a UI screen when emergency situation 2 occurs. By knowing about what situation occurred, we can infer what kind of emergency the person had.

4 Performance

4.1 Trajectory Tracking (Part 1)

We conducted 10 experiments on linear movement and 10 experiments on non-linear movement to measure the accuracy of our trajectory tracking algorithm. Table 1 shows the result of our trajectory tracking. In 20cm linear movement, we have 6.9812cm error with 4.1449cm standard deviation. Moreover, in 24cm non-linear movement, we have 6.7539cm error with 5.7743cm standard deviation.

4.2 Mobile Bodyguard (Part 2)

Part 2 of our project is a mobile guard. It is a service that detects a dangerous situation and sends a warning signal. Thus in this service, it is essential not to send an emergency alert in

	linear movement (Straight line 20cm)	
	Average error(cm)	Std(cm)
error(cm)	6.9812	4.1449
	non-linear movement (Curved line 22cm)	
	Average error(cm)	Std(cm)
error(cm)	6.7539	5.7743

Table 1: Result of Trajectory Tracking

		answer	
		shake	other gestures
result	shake	48(TP)	6(FP)
	other gestures	2(FN)	44(TN)

Table 2: Confusion matrix of Gesture Recognition

daily life and to send an emergency alert by recognizing a dangerous situation accurately. Therefore, we used a confusion matrix [4] to evaluate the model. Among them, we evaluated the performance of the model by focusing on accuracy. The reason is that accuracy is the ratio of true positive and true negative in all cases, so it is suitable for accurately recognizing dangerous situations in risky situations(true positive) and not recognizing them in daily life(true negative). Additionally, we evaluated the performance of our service how robust it is in everyday life.

Accuracy of Gesture recognition : To test how robust the model of our gesture recognition is for other gestures, we make shake gesture and other gestures 50 times respectively. The result of confusion matrix is shown in table 2. In this table, we can see that the Accuracy is 92%, Precision is 89% and Recall is 96%.

Accuracy of Voice recognition : To test how robust the model of our voice recognition is for other everyday voices, we make help voice and other voices 50 times respectively. The result of confusion matrix is shown in table 3. In this table we can see that the Accuracy is 93%, Precision is 97% and Recall is 88%.

Performance of Total System : Our mobile system operates based on gesture recognition and voice recognition. Therefore, the evaluation of each situation in situation 1 and situation 2 was omitted because it can be inferred through the evaluation of gesture recognition and voice recognition we explain before. To confirm the case of misrecognition of our mobile service in daily life, we make a testing scenario. We wear our device for 10 minutes and perform daily activities. As a result, emergency situations were not detected in daily life. So we can see that our emergency detection algorithm is robust enough in daily life.

		answer	
		help	other voices
result	help	44(TP)	1(FP)
	other voices	6(FN)	49(TN)

Table 3: Confusion matrix of Voice Recognition

5 Contribution

Usefulness. We were able to successfully detect two defined dangerous situations through mobile bodyguard. And, as a result of testing in daily life, it was confirmed that danger notifications were not sent in everyday life. Also, since it is implemented using the imu sensor existing in the smartwatch, the service can be implemented without additional equipment.

Novelty. This is the first service that allows smartwatches to detect and send alerts to dangerous situations that are not traditionally functions for convenience and health care.

References

[1] *Accelerometers*. https://www.hobbytronics.co.uk/accelerometer-info.

[2] *Arduino Nano 33 BLE Sense*. https://store-usa.arduino.cc/products/arduino-nano-33-ble-sense.

[3] *Bleak*. https://pypi.org/project/bleak/.

[4] *Confusion matrix*. https://en.wikipedia.org/wiki/Confusion_matrix.

[5] *TensorFlow Lite for Microcontrollers*. https://www.tensorflow.org/lite/microcontrollers.

[6] Colab. https://colab.research.google.com/.

[7] The Business Research Company. *Smart Watch Global Market Report 2021: COVID 19 Growth And Change to 2030*, May 2021. https://www.reportlinker.com/p06070190/Smart-Watch-Global-Market-Report-COVID-19-Growth-And-Change-to.html.

[8] Statista Research Department. *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025*, November 2016. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[9] Laura García, Lorena Parra, Miran Taha, and Jaime Lloret. System for detection of emergency situations in smart city environments employing smartphones. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 266–272. IEEE, 2018.

[10] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 74–77. IEEE, 2015.

[11] CrimeCriminal Justice Statistics. *Total number of crimes*, August 2021. https://www.crimestats.or.kr/portal/stat/directStatPage/T186633024325397.do.

[12] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.