UltraSafe: Ultrasound-based Driver Behavior Monitoring

Yatian Liu
University of Michigan, Ann Arbor
Ann Arbor, MI, USA
dougliu@umich.edu

Tianyang Shi
University of Michigan, Ann Arbor
Ann Arbor, MI, USA
tyshi@umich.edu

Mingke Wang
University of Michigan, Ann Arbor
Ann Arbor, MI, USA
mingkew@umich.edu

ABSTRACT

Driver safety has always been a significant concern. Driver fatigue and distraction are two of the principal risks of driver safety. They have caught more attention than before due to the recent development of assistive autonomous driving and increasing distraction from smartphone usage. These distracted driving and fatigued driving behaviors are accountable for more than half a million traffic accidents in the US every year. While the motor companies have their own driver monitoring systems, they are usually not very applicable due to low accuracy or privacy issues. In order to improve the driver safety but not interrupt the driver's privacy, we propose `UltraSafe`, an accurate, low-cost, and privacy-preserving driver monitoring system. By sending beamformed ultrasound signals and analyzing the reflected waves, `UltraSafe` can perform driver head monitoring with more than 95% classification accuracy without collecting the driver's personal identifiable information.

KEYWORDS

ultrasound, reflectometry, embedded systems, machine learning, driver monitoring

1 INTRODUCTION

Every year about 60,000 traffic accidents take place due to sleepiness-related problems in the US [5], and driver distraction is another main threat to driving safety [6]. Therefore, the driver's status is one of the critical factors influencing driving safety, and continuous efforts have been made to monitor drivers.

Traditional methods for driver monitoring can be broadly categorized into vision-based methods and non-vision-based methods [5]. Vision-based methods have been a hot research topic for a long time and are the most popular on-market solution for driver behavior monitoring [1, 4]. While it provides fine-grained driver behavior detection, it can pose severe privacy issues for the driver. Once the cameras were compromised, the personal information of the driver would be exploited and abused in a malicious fashion. non-vision-based methods are more heterogeneous and have different advantages and weaknesses, which will be discussed further in the related works section.

We propose `UltraSafe`, an ultrasound-based driver monitoring system inspired by previous non-vision-based monitoring methods that utilize radio frequency (RF) signals and ultrasonic sensing techniques applied to other fields. It is both non-intrusive for the driver and privacy-preserving compared to previous systems. By mounting a custom array of ultrasound transmitters and receivers on the overhead visor and sending various beamformed signal pattern, we can get rich information of the driver's posture from the reflected ultrasound signal. The collected data can be fed into a machine learning classifier such as a support vector machine (SVM) to determine driver's posture, and experiment results show close to 100% accuracy for single user cross validation and over 90% accuracy for cross user validation using a very small training dataset.

The main contributions of our work are:

(1) Applying ultrasonic sensing techniques to driver behavior monitoring and proving the feasibility.
(2) Building a hardware system that sends beamformed ultrasonic waves and receives reflected waves with a high sampling rate in real-time.

The organization of the report is as follows. Section 2 discusses previous works on driver behavior monitoring and ultrasonic sensing and our new system's relationship with them. Section 3 describes the system's design in detail. Section 4 shows the validation experiment setup and results. Section 5 discusses our current work's limitations and points some directions for future work. Section 6 concludes the report.

2 RELATED WORKS

2.1 Driver Behavior Monitoring

A general approach for driver behavior monitoring is applying computer vision. Several famous auto companies attempted to implement computer vision in their driver monitoring systems [1]. As early as 2006, Toyota mounted an IR camera on top of the steering column cover in the latest Lexus model to monitor the position and angle of the driver's head. Other commercial products such as Driver State Monitor (DSM) from Delphi also take computer vision as the main approach for driver behavior monitoring. DSM can analyze the driver's fatigue or distraction level by detecting and tracking the driver's facial features from a single camera directly in front of the driver. The camera is mounted on the dashboard with two IR illumination sources.

Non-vision-based methods usually involve tracking the driver's biological features or the vehicle's features [5, 7]. The biological signals, including electroencephalography (EEG), electrocardiogram (ECG), electro-oculography (EOG), and surface electromyogram (sEMG), are also used to help determine whether the driver is experiencing fatigue and drowsiness [1, 7], but the acquirement of these biological signals requires the driver to wear dedicated hardware. Drowsiness and distraction detection based on hand position on steering wheel, vehicle's lane deviation, and other vehicle movement patterns have also been studied, but many of these methods need complex augmentation of the vehicle and the correlation between the vehicle's movement and the driver's status is still debatable [5].

One new modality for non-vision-based methods is radio frequency (RF) sensing. Xie et. al propose a Wi-Fi-based driver monitoring system [8] for tracking driver's head movement. The system can track the movement within 4° to 10° using the Wi-Fi module on the smartphone, but due to the interference on 2.4 GHz channel and the changing interior environment, the performance maybe negatively affected.

2.2 Ultrasonic Reflectometry for Human Sensing

Reflectometry refers to object detection and characterization using reflection on waves. Sonar is a typical example of acoustic reflectometry, and ultrasonic reflectometry has been widely used in medical sensing and nondestructive structure testing. Reflected ultrasonic waves contain rich information, including but not limited to the time of flight (ToF), amplitude variations due to reflections, and multipath effects. Besides its traditional use, ultrasonic reflectometry is also suitable for human sensing. The emitted sound from daily sound sources has very low ultrasonic components, which means the signal-to-noise ratio (SNR) is high for ultrasonic signals. Also, human voice will not be collected by ultrasound receivers based on physical principles, and using ultrasound for human sensing will not collect personal identifiable information and therefore preserves privacy.

One recent work that applies ultrasonic reflectometry to human sensing is BeamBand [2]. This paper uses a wristband containing eight ultrasonic transducers to sense hand gestures. It achieves 94.6% accuracy for a six-class hand gesture set classification for one user, which shows that ultrasonic reflectometry can be used for complex human posture recognition. The design of our hardware system is partially based on that in [2].

3 SYSTEM DESIGN

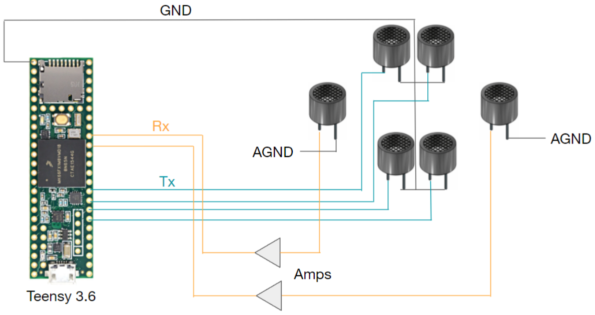


Figure 1: System Schematic

The UltraSafe system is composed of one MCU, four ultrasound transmitters, and two ultrasound receivers. Inverting amplifiers built from op-amps are used to amplify the weak signal from receivers. Since our design has a high demand for timing performance, we chose Teensy 3.6 as MCU. It uses an ARM Cortex-M4 CPU and has a 16 MHz crystal for the system clock and a phase locked loop that increases the 16 MHz up to the 180 MHz system clock speed. It can use the Arduino library and many other third-party software libraries, which facilitates embedded software development. It is also equipped with a Programmable Delay Block (PDB) as stable delay timer. UT(R)-1640K-TT-2-R are used as ultrasonic transmitters(receivers). These transducers have a center frequency of 40 kHz and a bandwidth of 2000 Hz. The system schema is shown in Fig. 1.

3.1 Hardware Design

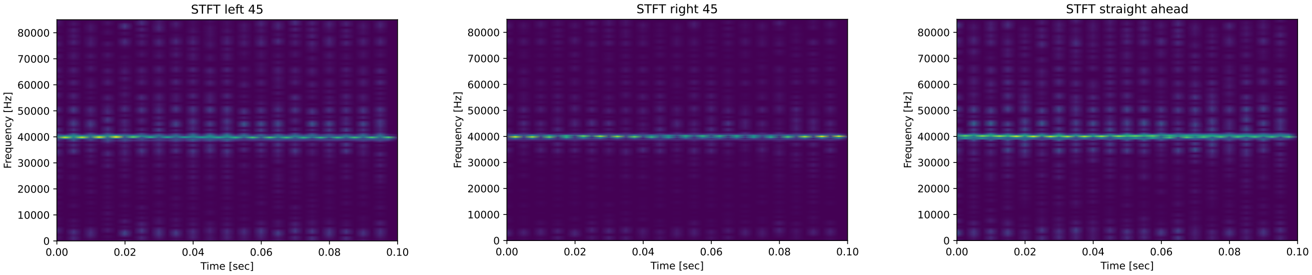
Both driver signals for transmitters and the reflected ultrasonic signals from receivers are managed by GPIO on Teensy. To eliminate the interference of power rails between transmitters and receivers, we use two isolated ground signal lines for transmitters and receivers. The four transmitters share the GND of Teensy while the two receivers and op-amp circuits take the AGND. In addition, multiple capacitors are used to ensure power integrity and reduce noise.

During the preliminary experiment, we found the amplitude of signal directly from receivers was lower than 200 mV, while the range of ADC was 0-3.3 V. Thus, we applied op-amp in our design to achieve a wider signal amplitude range. We chose AD8032ANZ as our op-amp and built an inverting amplifier circuit to achieve a gain of around 27 dB and a DC biasing of around 1.65 V is applied to center the signal in the ADC range. The result of our preliminary experiment shows that the maximum peak-to-peak amplitude of received signal is around 1.2 V.

3.2 DMA for Rx and Tx

The hardware system needs to handle fast ADC sampling, precise GPIO output timing control for beamforming, fast serial port transmission, ad other auxiliary tasks concurrently. Running all the tasks on the CPU will make them interfere with each other and cause unstable timing, so we choose to offload CPU work using the direct memory access (DMA). Only the ADC sampling task and the GPIO output timing control task are done using DMA since the other tasks do not have strict real-time timing constraints.

Output data of the two ADCs are directly written into two memory buffers, respectively. Sampling will stop after the buffers are full, and a variable will be set to let the CPU know this information and start the next round of sampling. Alternatively, the double buffer can be used for one ADC, and when one buffer is full, the ADC will write to the other buffer, and which allows continuous sampling. The continuous sampling mode is only used for DMA functionality testing in Section 4.1.1 since frame-based sampling and transmission is more suitable for the actual system. To set the Tx GPIO outputs using DMA, we choose four GPIO ports that are associated with the same hardware register, and write to this register periodically according to look-up tables (LUT) stored in memory. One LUT corresponds to one period of the 40 kHz ultrasonic signal, which is 25 µs. The LUT has a length of 100 and the programmable delay block (PDB) on Teensy is used to trigger GPIO writes with a period equal to the wave period (25 µs) divided by LUT length (100). For each data point in an LUT the four bits in the register that corresponding to the four Tx outputs are set according to the delay needed for beamforming. One LUT is created for each beamforming angle, and switching beamforming angles is done by switch LUTs for DMA. The minimum delay between two Tx output is equal to the PDB period, which is 0.25 µs. This precision is good enough for sound waves in the air since 0.25 µs of delay corresponds to a tiny 85.7 µm travel distance.



(a) Gesture: left 45°

(b) Gesture: right 45°

(c) Gesture: Straight Ahead

Figure 2: Frequency Domain Information

3.3 Sample Frame Design

The beamformed signal is chosen to transmit pulses instead of transmitting continuously in order to record time-of-flight (ToF) information. We first start a round of ADC sampling that lasts for 5 ms, and then immediately begin transmitting the beamformed signal for 0.5 ms, leaving around 4.5 ms of idle time. Interrupts are turned off during the initiation of ADC sampling and Tx signal transmission to stabilize timing. ADC sampling starts before signal transmission so that the starting signal will always be collected. The signal transmission time and ADC sampling time is chosen so that enough sampling time is reserved to get the reflected ultrasonic wave and the rate for getting a full frame of samples is high enough.

After each round of sampling, the data from the two ADCs are packed with headers to form a data frame. The frame of data is transmitted to the host computer via a USB serial port, and then the LUT for beamforming is switched to the next beamforming angle table to prepare for the next round of sampling. Taking data transmission and other overheads into account, collecting and transmitting one frame of data takes about 7.5 ms according to experimentation, which corresponds to about 133 frames per second.

3.4 Beamforming

Beamforming is a signal processing technique to improve the SNR of the received signal, eliminate the undesirable interference source, and focus transmitted signals to specific locations. It is widely used in radar, sonar, medical imaging, and audio applications. The beamforming technique can be used to focus the transmitted signal of a sensor array to a specific direction.

`UltraSafe` adopts the beamforming technique to focus the signal of the ultrasound transmitter to a set of different angles to get a diverse response for the reflected signal.

Our transmitter has a center resonance frequency of 40 kHz, so we will use the square wave with frequency 40 kHz and a duty cycle of 50% to power the transmitter.

For an array of narrowband signals, beamforming can be expressed by multiplying the signal with a complex exponential with an appropriate phase shift.

Suppose the each ultrasound transmitter emit the signal:

$$E_n = A_n \sin(\omega t + n\Phi) \tag{1}$$

where the A_i is the signal's amplitude, ω is the angular speed, Φ is the phase of the signal.

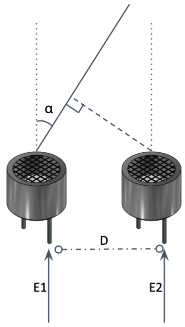


Figure 3: Beamforming

Φ is used to depict the phase shift due to both (a) travel distance d, (b) observation angle θ, and (c) phase offset ϕ.

$$\Phi = \frac{2\pi}{\lambda}\tau + \phi = \frac{2\pi}{\lambda} d \sin\theta + \phi \tag{2}$$

The overall observed signal is

$$E = \sum_{n=0}^{N-1} E_n = A_0 \sin\omega t + \cdots + a_{n-1} \sin(\omega t + (N-1)\Phi) \tag{3}$$

The parameter ϕ can be selected accordingly to focus the signal to the desired focus direction. If we want the signal to steer to angle α, the phase shift added to each transmitter can be expressed as:

$$\phi = \frac{2\pi}{\lambda} d \sin\alpha \tag{4}$$

where $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, d is the distance between two consecutive transmitter, and λ is the wave length of the signal.

In practice, instead of applying the phase to the signal, the phase shift is usually converted to the time delay.

$$\Delta t_\phi = \frac{\phi}{2\pi f} = \frac{d \sin\alpha}{\lambda f} = \frac{d \sin\alpha}{v} \tag{5}$$

where the f is the frequency of the wave, v is the velocity of the wave (340 m/s).

By calculating the lookup table (LUT) for each defined focus angle, we can modulate the input signal of the transmitters efficiently on the fly.

Choice of beamforming angle. UltraSafe consists of four transmitters and can perform 3D beamforming. The more beamforming angle we choose, the more information we can get to infer the gesture of the driver. But if we choose too many angles, it will take too much time to collect the reflected signals, thus decreasing the detection frequency. In the experiment, we choose 5 focus angles, $(-10°, -5°, 0°, 5°, 10°)$.

3.5 Short-time Fourier transform (STFT)

The transmitted signal will be reflected multiple times before it finally reaches the receiver side. Because the driver is moving all the time, the propagation path of the signal from the transmitter to the receiver is also changing. These changes generated by the driver's movement will cause a shift in the frequency of the reflected signal. This phenomenon is called the Doppler effect. The observed frequency of the signal can be expressed as:

$$f' = \frac{v \pm v_0}{v \pm v_s} f \tag{6}$$

where f' is the observed frequency, the v_0 and v_s denote the velocity of the observer and the source relative to the medium respectively, v and f is the velocity and frequency of the signal.

STFT is a commonly used method to capture the changes of frequency component over time, and by applying STFT to the signal, we can evaluate the Doppler effect caused by the driver movement and further help us understand the gesture of the driver.

The STFT is carried out using the following procedures:

(1) Break the data sequence into multiple chunks.
(2) perform fat fourier transform to each chunks.
(3) Add all the fourier transform result together:

$$\mathrm{STFT}(x[n])(m, \omega) = \sum_{n=-\infty}^{+\infty} x[n]x[n-m]e^{-j\omega n}$$

3.6 Classification Algorithm Design

As discussed in the previous part, UltraSafe will collect the signal from both ultrasound receivers in the form of frames. The collected signal can be interpreted and utilized in both the time and frequency domain. We found that the time domain contains much more information than the frequency domain. As shown in Fig. 5, time domain information shows the vast difference with different gestures, different beamforming angles, and different receiver positions. While in the frequency domain (shown in Fig. 2, all the signal are centered at $f = 40$ kHz and doesn't exhibit much difference. This is because (a) the receivers have a sharp frequency response at $f = 40$ kHz, (b) the transmitters are sending directional signals, which will greatly suppress multipath effects.

Based on the previous analysis, in UltraSafe we only use the time domain information as features for classification. To validate the effectiveness of the time domain information, we use t-SNE to visualize the data. As shown in Fig. 4, each color corresponds to the data of one gesture. Despite some overlap, most gestures can be distinguished from other gestures, suggesting that the time domain information alone is adequate to determine the gestures.

As for the classifier, we choose support vector machine (SVM) because of its stable performance across multiple classification tasks [3]. In Sec. 4.3, we will show that SVM is able to provide high accuracy classification.

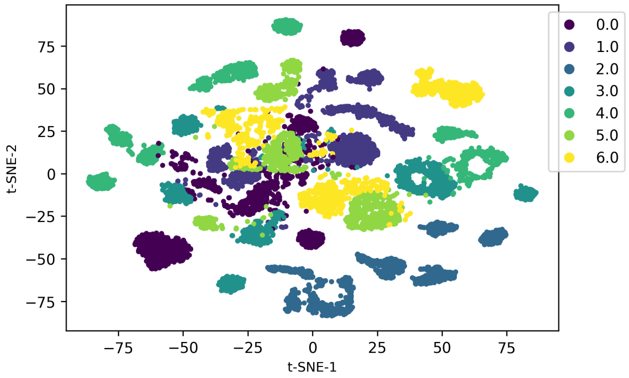


Figure 4: t-SNE plot of time domain signals

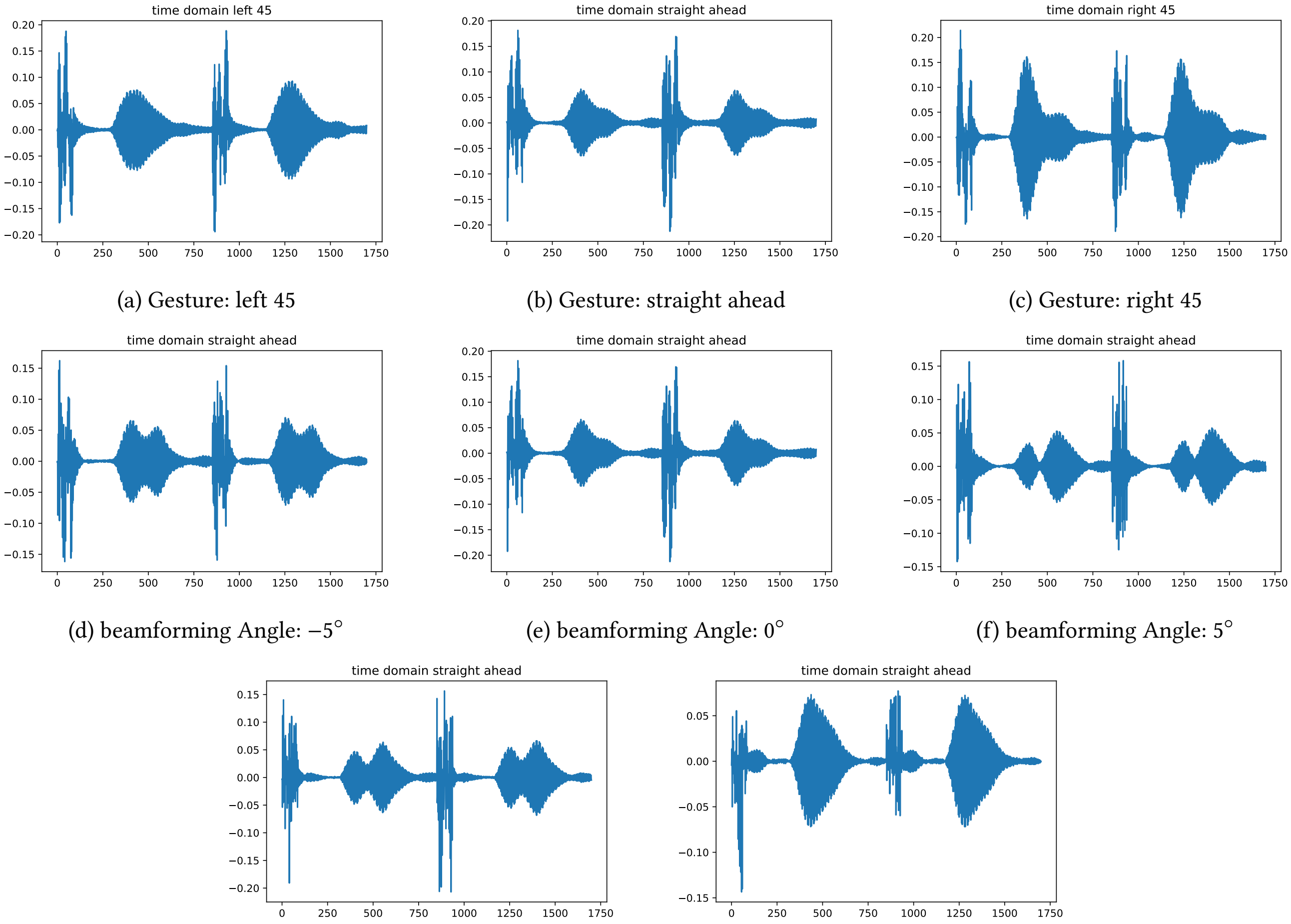
4 EVALUATION

4.1 Preliminary Verification

4.1.1 Verification of ADC DMA sampling. To verify that the DMA setup for ADC is working, a sine wave from a signal generator is used as input for one ADC, and the ADC DMA is set to continuous sampling mode with double buffer. The sampled voltage data is transmitted to the host computer and then being scaled and plotted as a time series. The plot shows a continuous sine wave whose frequency and amplitude matches the settings of the signal generator, which proves that the ADC DMA is working correctly.

4.1.2 Verification of beamforming delay control. Our goal in this part is to generate four tracks of 40 kHz square wave signal with 3 µs phase shift. The phase shift is measured by an oscilloscope. This is to verify that we can control the time difference of transmitter driving signals precisely to realize beamforming. We tried several approaches to generate the signal, including multiple timers with system delay and using single timer to simulate system ticks in RTOS. Finally we implemented a look-up table and use DMA and PDB of Teensy to manage the output on each GPIO. Compare to the former approaches, the frequency and phase shift generated by LUT and PDB were more accurate.

4.1.3 Verification of signal amplification. In order to achieve a wider signal amplitude range, we chose AD8032ANZ as our op-amp and built an inverting amplifier circuit. We implement an amplifier circuit of about 27 dB gain. In the first step of verification, we input a 40 kHz, 50 mV sinusoidal signal to simulate the signal from the ultrasound receiver. The output showed a steady amplitude of 1.2 V, so we move on to use ultrasound receiver as input source to simulate real case. The result of this experiment showed that the amplified receiver signal has desired amplitude and could be sampled by ADC precisely. So far all the functions had been verified and we were ready to move on to data collection.



(a) Gesture: left 45

(b) Gesture: straight ahead

(c) Gesture: right 45

(d) beamforming Angle: -5°

(e) beamforming Angle: 0°

(f) beamforming Angle: 5°

(g) left Rx

(h) right Rx

Figure 5: Time Domain Information

4.2 Experiment Setup

We perform extensive data collection of the 7 driver postures, namely looking straight ahead, looking down, looking up, looking left 45 degrees, looking left 90 degrees, looking right 45 degrees, looking right 90 degrees.

The data is collected from three different testers. For each tester, we collect four rounds of 5-second data for each posture.

4.3 Experiment Results

4.3.1 Time Domain Information. As shown in Fig. 5, the time domain information has the following characteristics:

(1) Different gestures have signal envelopes with different amplitudes.
(2) Different beamforming patterns will cause different time responses.
(3) Different receivers will pick up different time responses.

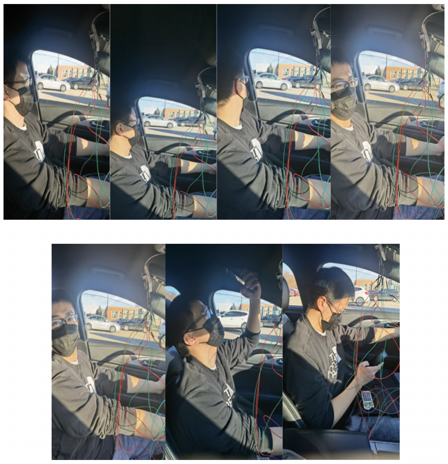


Figure 6: 7 driver postures

These time response signals show that using the current frame design, UltraSafe can provide abundant time domain information to identify different gestures.

4.4 Overall Cross Validation Performance

In this section, we will mainly focus on the classifier's performance when the data from all three testers are used for training and testing.

We first investigate the ability to identify the different gestures. We used all the data collected from the three testers and divided 70% into the training set and 30% into the test set.

In the training stage, we use fivefold cross-validation, which is splitting the training set into five sets of equal size, train the model on four sets and validate the performance on the remaining one set. The model that achieves the best performance on every validation set will be adopted.

As shown in Table. 1 shows the performance on the test set. The SVM model with parameter $C = 10, \gamma = 0.1$, and kernel = radial basis function (RBF) achieved an average F1-score of almost 100%. This high accuracy aligns with the previous t-SNE graph and further proves that the time domain signal contains enough information to determine the driver's gesture.

Gesture	Precision	Recall	F1-score	Support
straight ahead	1.00	0.99	1.00	364
look up	0.99	0.99	0.99	348
look down	1.00	0.99	1.00	376
left 90°	0.99	0.99	1.00	349
left 45°	1.00	0.99	1.00	360
right 45°	1.00	0.99	1.00	364
right 90°	0.99	0.99	0.99	356
accuracy			1.00	2508
macro avg	1.00	1.00	1.00	2508
weighted avg	1.00	1.00	1.00	2508

Table 1: Classification Result

4.5 Cross-User Validation

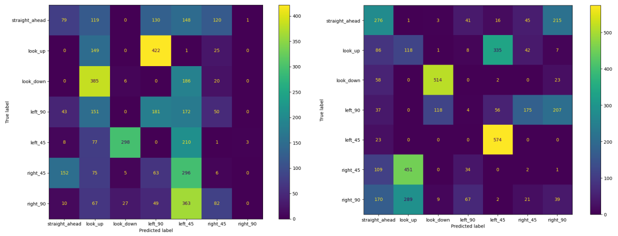
Compared to classification performance on the collected data, we should focus more on how the model performs on new users.

We conduct two experiments to examine the cross-user performance of the model: (a) train on one driver, validate on one driver; (b) train on two drivers, validate on one driver. We will abbreviate them as "train 1 validate 1" and "train 2 validate 1" in the following discussion.

As shown in Fig. 7, the train 1 validate 1 method has a poor performance and almost all the data has been mis-classified. But after we use the train 2 validate 1 method, although the performance is still not near 100%, the F-1 score has been greatly improved for the look down and left 45° gesture.

This inconsistency between the cross-user validation and the former experiment is caused by the lack of data diversity from different divers. In the train 1 validate 1 experiment, the model is not able to generalize the pattern between different drivers, while in the train 2 validate 1 experiment, the model can figure out some common features between different users, and that's why the performance is better than the train 1 validate 1 experiment.

According to this phenomenon, we can conclude that UltraSafe can achieve better cross-user validate performance with more diverse user data.



(a) train 1, validate 1

(b) train 2, validate 1

Figure 7: Cross-user Validation

5 DISCUSSION AND FUTURE WORKS

In our evaluation section, we have shown that UltraSafe can be used to determine the driver's head position accurately. However, the UltraSafe can be further improved.

First, as shown in Sec. 4.5, our training data is not diverse in the sense of the different types of driver figures (height, weight, age). By collecting and training on more diverse driver data, UltraSafe will be able to perform higher accuracy movement detection for every new user.

Second, we can only use two receivers due to the limitation of the MCU we use (the Teensy 3.6). Teensy 3.6 only contains two built-in ADCs, and if we were using more receivers, we would have to use the different channels of the ADC, which will lower the sampling rate, making it challenging to meet the Nyquist sampling frequency for the 40 kHz ultrasonic waves.

Finally, fine-grained data collection should be carried out to validate the potential of using UltraSafe for driver gesture tracking. Our data is labeled according to 7 different gestures. The driver head movement monitoring is treated as a classification problem. But according to the high accuracy classification performance, we believed that UltraSafe has potential in performing continuous head tracking using regression. In the future, we will collect the ground truth head movement using IMU, and evaluate the performance of head movement tracking using UltraSafe.

6 CONCLUSION

In this project, we present UltraSafe, an accurate, privacy-preserving driver behavior monitoring system. By exploiting the latent information embedded in the reflected waves of the beamformed signal, UltraSafe can perform high accuracy driver head movement monitoring while preserving driver's privacy.

REFERENCES

[1] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. 2011. Driver inattention monitoring system for intelligent vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems* 12, 2 (2011), 596–614. https://doi.org/10.1109/TITS.2010.2092770

[2] Yasha Iravantchi, Mayank Goel, and Chris Harrison. 2019. BeamBand: Hand gesture sensing with ultrasonic beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.

[3] Jekyll. 2021. Random forest vs SVM. https://www.thekerneltrip.com/statistics/random-forest-vs-svm/.

[4] Hang-Bong Kang. 2013. Various Approaches for Driver and Driving Behavior Monitoring: A Review. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

[5] Sinan Kaplan, Mehmet Amac Guvensan, Ali Gokhan Yavuz, and Yasin Karalurt. 2015. Driver behavior analysis for safe driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 16, 6 (2015), 3017–3032.

[6] Michael A Regan, John D Lee, and Kristie Young. 2008. *Driver distraction: Theory, effects, and mitigation*. CRC press, Chapter Defining driver distraction.

[7] Gulbadan Sikander and Shahzad Anwar. 2018. Driver fatigue detection systems: A review. *IEEE Transactions on Intelligent Transportation Systems* 20, 6 (2018), 2339–2352.

[8] Xiufeng Xie, Kang G. Shin, Hamed Yousefi, and Suining He. 2018. Wireless CSI-based head tracking in the driver seat. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. ACM, New York, NY, USA, 112–125. https://doi.org/10.1145/3281411.3281414