Static Hand Gesture Recognition

Nathan Nakkapalli, Yufeiyang Gao, Dominik Dulak, and Rachel Boeke

ABSTRACT

Users typically interact with a device through direct contact or with a remote controller. However, these interfaces do not provide an intuitive, natural experience for the user. In this work, we attempt to identify static gestures in real-time to provide users with a more natural user interface. We created a wearable device containing a VPU and transducer in contact with the user's wrist. The transducer sends out ultrasonic waves through a function generator. The waves go through the user's hand and are received by the VPU as a reflection. Different gestures yield different reflections. We used this difference to train the data collected by the VPU on an offline machine learning algorithm and imported the final model into T4Train [1], a real-time machine learning software. Using T4Train, we were able to recognize three different gestures (open hand, closed hand, and two finger "ok" sign) in real-time. Limitations of this gesture recognition were the need to train a model specific to each user and the need to maintain identical training and testing conditions. Future work to address these limitations includes training a model on a larger number of samples collected under diverse conditions and improving real-time prediction speed by reducing the duration of each sample. With these improvements, static gestures may be a valuable addition to the interfaces used to control presentations, play and pause media, play games, and more.

INTRODUCTION

Gestures are widely used for communication in the real world. More than 70 million people worldwide use sign language. This project was first motivated by the idea of interpreting ASL to eliminate the barrier between those who use ASL and those who do not. There are more than 4500 ASL sign gestures. It was not feasible to identify this large number of gestures in 1 month, so we investigated applications requiring fewer gestures.

The traditional user interface, including components like the keyboard, mouse, touch screen, or remote controller, requires physical interaction and cannot be customized for individuals. Current user interfaces are not intuitive, and this poses a barrier for people who are not familiar with technology. Moreover, due to the inability to provide customized interaction between users and the device, current interfaces also pose difficulties for people with disabilities. Our project aims to provide a more intuitive way to interact with one's device, such as controlling slides with closed-hand and open-hand gestures, scrolling an article with a thump-up or thumb-down, and picking/hanging up a call with crossed fingers. Through static hand gesture recognition, users can customize their experience by choosing gestures that are the easiest to perform based on their backgrounds.

RELATED WORK

We learned from multiple existing works as we worked on this project, namely Echowrist [2], SAWSense [3], and Interferi [4].

Echowrist estimates hand pose and hand-object interaction with a microphone and speaker placed on each side of the wrist. Because they have a microphone and speaker on each side of the wrist, they can calculate the difference between transmitted and received acoustic data. Their machine learning model is also trained on this difference. However, because they are using a microphone, which is sensitive to background noise, their application is limited to quiet locations.

SAWSense uses a Voice Pick Up (VPU) sensor to collect surface acoustic waves (SAWs) generated when the user's hand engages in touch-based activities. The authors then trained a machine learning model which was able to recognize user interaction with daily objects. Specifically, the authors were able to classify gesture inputs (like dragging a finger over a desk or tapping on the desk) and conduct activity recognition (like whisking in a bowl versus chopping food on a cutting board).

Finally, Interferi tracks hand and face movement with transducers emitting and receiving acoustic signals. By utilizing multiple transducers, the authors were able to do continuous tracking of the hand, wrist angle, and smile intensity of a person's face.

Our project combines hardware from SAWSense and Interferi in that we use both a VPU and transducer. As SAWSense is only using a VPU, the application is limited to dynamic gestures that make sounds (or more specifically, activities that produce SAWs). By using one VPU and one transmitting transducer (see Figure 2), which is less hardware than used in Interferi, our project is able to recognize static gestures (gestures that hold state and are not necessarily emitting SAWs). Compared to Echowrist, our project uses a VPU, which is less sensitive to environmental noise than a traditional microphone, allowing for a wider range of applications than Echowrist. However, since Echowrist uses signal receiving and transmitting devices on both sides of the wrist, it has more distinguishable data of signals going through different gestures. A future extension of this work inspired by Echowrist could address this issue by placing one VPU on each side of the wrist, one measuring transmitted signal and one measuring the received signal.

SYSTEM OVERVIEW

Our approach consisted of three main parts: Hardware, Offline Machine Learning, and Real Time Machine Learning.

Hardware Overview

We use a transducer (with a bandwidth of 25khz +/- 2) with a Voice Pickup Unit to capture Surface Acoustic Waves. The transducer and the VPU are both placed in direct contact with the skin. This allows the transducer to produce SAWs which are then picked up by the VPU. The transducer is driven using a function generator with a frequency sweep from 23 to 27 kHz with a period of 0.1s carrying a square wave at 20 Volts Peak-to-Peak and 10% duty cycle. We decided on this waveform through empirical testing with different factorizations. This waveform showed promise with visually distinguishable FFT features and high accuracy on test data for our machine learning models.

We utilized a VPU over a standard microphone because it is not as susceptible to environmental noise compared to a traditional microphone. An ESP32s3, is powered by a portable USB charger, is used to read data from the VPU using an wired I2S communication protocol and that data is streamed to the laptop using Wifi. We use another ESP32 to create a private Wifi network, allowing both the laptop and the ESP32s3 to send and receive data when connected to the private Wifi.

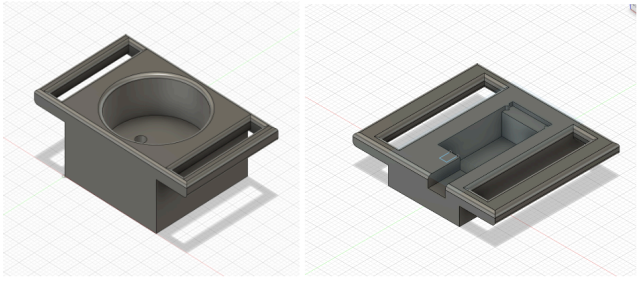

Figure 1: Custom 3D Printed housings for VPU & Transducer

In order to secure the VPU and transducer to a human wrist using velcro straps, we designed and 3D printed custom mounts for the VPU and transducer (see Figure 1). They include loops (slits on the sides) to adjust the size of the velcro strap, creating a one-size-fits-all wearable that can be used on anyone's wrist. Edges for the strap were rounded to allow the strap to be adjusted quickly and to minimize discomfort to the user from any sharp edges.

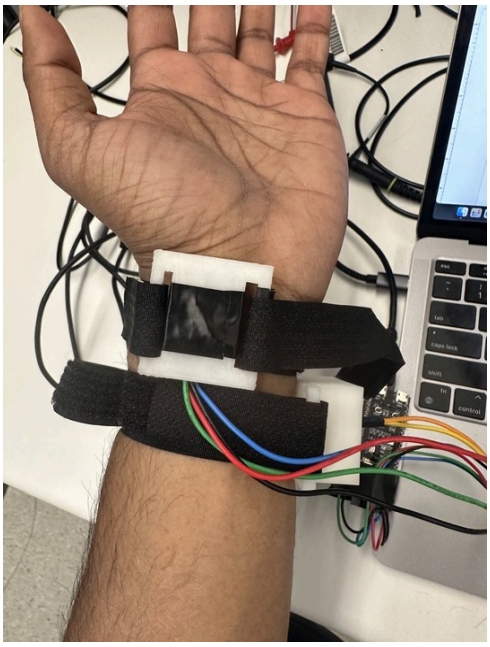

Figure 2: Our VPU (top) and transducer (bottom right) attached to the underside of the wrist as a wearable

Our current setup includes two separate wrist wearables (see Figure 2). This allowed us to increase the distance between the VPU and the transducer and resulted in more distinguishable data per gesture.

Offline Machine Learning Overview

We initially used a 200 millisecond pulse and MFCC featurization to train a ML model. This setup performed very poorly, rarely exceeding 50% training and testing accuracy. To address this low accuracy, we experimented with both the underlying signal and featurization method.

In our final model, as stated previously, we drove the transducer using a frequency sweep from 23 to 27 kHz with a period of 0.1s carrying a square wave at 20 Vpp and 10% duty cycle.

Using this signal and the data that the VPU collected, we featurized the data using a Fast Fourier Transform (FFT). We then trained on a number of models and compared accuracies to identify the best model to use in the real time detection. Nearly all of the models exhibited greater than 97% accuracy on training data.

There is actually a specific reason for using FFT featurization. When doing experiments, we could clearly see that there were distinct but constant FFT featurization states that corresponded to a specific gesture state. In other words, we could visually verify that a given static gesture directly corresponds to a unique FFT output that holds steady over time. We knew these FFTs were unique for each gesture as the maximum magnitude within the FFT was different for each gesture (but of course, the FFT output would remain relatively the same for a particular gesture as

time went on, even when we switched between different static gestures). This also proves that our model was not learning off of random noise, since when we turned off the function generator, the FFT output was not steady or constant but rather continuously changing. Moreover, since we could visually differentiate between different gestures based on the FFT output, it makes sense that machine learning is doing exceptionally well during training.

Real Time Machine Learning Overview

Once we obtained high accuracy offline results, we transformed these offline machine learning models into real time models. To this end, a program called T4Train was repurposed. T4Train is a program which was specifically designed for real time machine learning and abstracts away the challenges of real time inferencing (classification). This program was then adapted to remove bugs, retrieve VPU data via Wifi, and utilize our pre-trained models from offline training. The inference speed is dependent on the number of frames (amount of data) that is collected before a prediction will be made. The amount of data that is collected before a prediction is the same amount of data we used when collecting training data. That is, if we collected 0.5 seconds samples of training data, then we would also collect 0.5 seconds of data before making a real time prediction. There is a potential tradeoff between inference speed and accuracy. Increasing the inference speed could mean less data will be collected before making a prediction. Thus, the model may give an incorrect prediction before arriving at the correct prediction. Future work can consider how to increase inference speed by further analyzing static gestures and how changing from one static gesture to another static gesture can affect data collection and inference.

STUDY DESIGN

We used three gestures for testing: open, closed, and two finger, shown in Figure 3. In order to obtain the best results, participants were tested with custom fit models. That is, they had to first train our machine learning model before we conducted testing soon after.

Figure 3: Closed, open, & two finger gestures

For each participant, we recorded at least 20 samples (around 0.5 second recording per sample) for each gesture. This data was specific to the participant – the data collected was with the wearable VPU (sampling at 96 kHz) and transducer on their wrist. We then trained a model using the data collected for that participant. The trained model was then uploaded to T4Train and used to classify gestures in real time during testing.

We tested the performance of each participant's model by telling the participant to form a gesture and recording the prediction shown in T4Train. We then evaluated whether the prediction was correct. Each participant was given 10 gesture cues. 5 users participated in the study.

Training and testing were done under identical conditions. The participant held their arm in the same position and remained sitting in the same spot between training and testing. Additionally, participants could provide any optional feedback they had regarding the user experience.

RESULTS

After testing with 5 participants over a total of 50 real-time trials, we obtained the following confusion matrix (Figure 4).

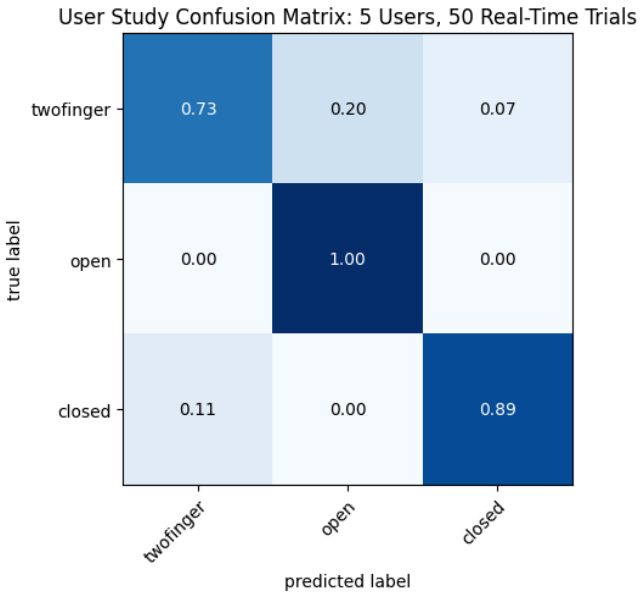

Figure 4: User study confusion matrix

Open was the most accurate gesture with 100% accuracy. Two finger and closed were occasionally confused, with 11% of two finger gestures misidentified as closed and 7% of closed gestures misidentified as two finger.

For real time predictions, we used the Support Vector Classifier (SVC) model as it produced at least 98% offline training accuracy in identifying the three gestures for each participant.

During testing, we observed a small delay between forming a gesture and seeing the correct prediction. In the future, this might be corrected by reducing the duration of training and testing samples (currently set at 0.5 seconds per

sample). Further testing is needed to see if reducing the sample duration will affect model accuracy.

In terms of participant feedback on the user experience, one participant stated she really liked the 3D housings for the VPU and transducer because they felt comfortable. Another person felt that the ideal positioning of the VPU and the transducer was too picky or sensitive to minor changes.

During training and testing, we analyzed FFTs for each user to ensure we could distinguish between gestures. For two random users, the FFT outputs for each sample are graphed below (thicker line means overlap between samples and thinner line means less overlap). Different gestures appeared to create the same shapes among different users, but the main FFT features (within the 20 to 40 kHz band) had different magnitude levels for different users (see Figure 5). Within each user, we can see that the gap or difference in magnitude of the two finger gesture compared to either the closed or open gesture is relatively small. On the other hand, the magnitude difference is quite large and apparent between open and closed gestures. This can explain why the real time model had some difficulties predicting between two finger and closed/open gestures versus open gesture and closed gesture.

It's also important to note three gestures produced a unique set of FFT features for only one specific user, which means we must retrain for each specific user. Also, the real time accuracies are lower than the offline model accuracies. Perhaps the offline models are overfitting (since we are obtaining extremely high training accuracies) and thus are not classifying as well as they could in real time. Future work can investigate increasing real time accuracy by applying regularization techniques to offline models to decrease model overfitting and/or find better ways to increase the gaps between each gesture's respective FFT output features (making it easier for real time classification to predict correctly).

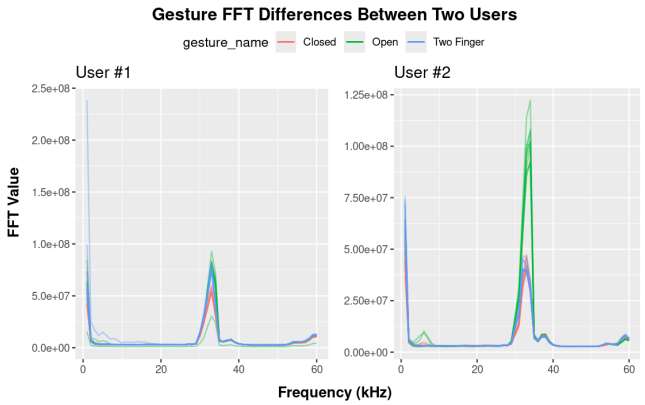

Figure 5: Sample FFTs for two users

CONCLUSION

In this project, we aimed to provide a more intuitive, natural way to interact with devices using real-time static gesture recognition. Our wearable final product consisting of a VPU, transducer, and ESP32 is portable, lightweight, and not impacted by environmental noise. We achieved high test accuracy offline using FFT featurization and an SVC model. Transforming this model into real-time classification proved more challenging, but we were able to identify open gestures with accuracy 100%, closed gestures with accuracy 89%, and two finger gestures with accuracy 73% during a 50-trial test with 5 users.

There are several limitations of our final product. First, to get the best accuracy, a user must use a model trained specifically for them. More training data needs to be collected in the future from a larger number of people and under different conditions to create a more robust model. Second, these models are very sensitive to placement of the VPU and transducer wristbands. Again, collecting more training data can resolve this issue. Third, real-time prediction speed is somewhat slow. Reducing the sample duration may improve real-time prediction speed, but may also decrease model accuracy. Fourth, since we only used one VPU and transducer, adding an additional VPU or transducer can provide more data and yield better classification accuracy. Finally, there is room for improvement on machine learning and featurization methods. We can continue to experiment with different underlying signals from the function generator, machine learning models, and featurization methods to improve prediction accuracy and expand the number of gestures we can identify with high accuracy.

REFERENCES

1. https://github.com/ISC-Lab/T4Train
2. Zhang, R., Agarwal, D., Yu, T. C., Gunda, V., Lopez, O., Kim, J., Yin, S., Dong, B., Li, K., Sakashita, M., Guimbretiere, F., & Zhang, C. (2024). EchoWrist: Continuous hand pose tracking and hand-object interaction recognition using low-power active acoustic sensing on a wristband. Proceedings of the [Human-Computer Interaction]
3. Iravantchi, Y., Zhao, Y., Kin, K., & Sample, A. P. (2023). SAWSense: Using Surface Acoustic Waves for Surface-Bound Event Recognition. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). Association for Computing Machinery, New York, NY, USA.
4. Iravantchi, Y., Zhang, Y., Bernitsas, E., Goel, M., & Harrison, C. (2019). Interferi: Gesture sensing using on-body acoustic interferometry,
5. Human-Computer Interaction