This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2962760, IEEE Transactions on Mobile Computing

Towards Low-cost Sign Language Gesture Recognition Leveraging Wearables

Tianming Zhao*, Jian Liu†, Yan Wang*, Hongbo Liu‡ and Yingying Chen†
*Temple University, Philadelphia, PA 19122 tum94362,y.wang@temple.edu
†Rutgers University, Piscataway, NJ 08854 jianliu@winlab.rutgers.edu,yingche@scarletmail.rutgers.edu
‡Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202 hl45@iupui.edu

Abstract—Different from traditional gestures, sign language gestures involve a lot of finger-level gestures without wrist or arm movements. They are hard to detect using existing motion sensors-based approaches. We introduce the first low-cost sign language gesture recognition system that can differentiate fine-grained finger movements using the Photoplethysmography (PPG) and motion sensors in commodity wearables. By leveraging the motion artifacts in PPG, our system can accurately recognize sign language gestures when there are large body movements, which cannot be handled by the traditional motion sensor-based approaches. We further explore the feasibility of using both PPG and motion sensors in wearables to improve the sign language gesture recognition accuracy when there are limited body movements. We develop a gradient boost tree (GBT) model and deep neural network-based model (i.e., ResNet) for classification. The transfer learning technique is applied to ResNet-based model to reduce the training effort. We develop a prototype using low-cost PPG and motions sensors and conduct extensive experiments and collect over 7000 gestures from 10 adults in the static and body-motion scenarios. Results demonstrate that our system can differentiate nine finger-level gestures from the American Sign Language with an average recognition accuracy over 98%.

Index Terms—Sign Language Translation, Photoplethysmography (PPG), Wearables, Human-Computer Interaction (HCI)

1 Introduction

The popularity of wrist-worn wearable devices has a sharp increase since 2015, an estimation of 101.4 million wrist-worn wearable devices will be shipped worldwide in 2019 [1]. Such increasing popularity of wrist-worn wearables creates a unique opportunity of using various sensing modalities in wearables for pervasive hand or finger gesture recognition. Hand and finger gestures usually have diverse combinations, which present rich information that can facilitate many complicated human computer interaction (HCI) applications. For example, wearable controls, virtual reality (VR)/augmented reality (AR), and automatic sign language translation. Taking the automatic sign language translation as an example illustrated in Figure 1, a wrist-worn wearable device (e.g., a smartwatch or a wristband) could leverage its sensors to realize and convert sign language into audio and text and back again, which will greatly help people who are deaf or have difficulty hearing to communicate with those who do not know the sign language. Recently, Er-Rady *et al.* [2] provide a review of the existing immature automatic sign language translation methods, which motivates us to develop a robust finger-level gesture recognition system to help solve the problem.

Existing solutions of gesture recognition mainly rely on cameras [3], [4], [5] microphones [6], [7], radio frequency (RF) [8], [9], [10] or special body sensors (e.g., Electromyography (SEMG) [11], Electrical Impedance Tomography (EIT) sensor [12], and electrocardiogram (ECG) sensor [13]). The approaches using cameras face occlusion and privacy issues. Microphones are vulnerable to ambient acoustic noises. The RF-based approaches are usually known to be device-free, but they are very sensitive to indoor multipath effects or RF interference. Using special body sensors for gesture recognition is more robust to environmental noises but requires extra cost and manpower of installation. Recently, motion sensors in wearables present their great potential in hand and finger gesture recognition on the wrist [14], [15], but motion sensors are sensitive to body motions, which makes them difficult to identify fine-grained finger-level gestures, such as sign language gestures. Recently, a few PPG-based gesture recognition work [16], [17] have been proposed, but they mainly focus on recognizing whole-body human activities such as standing, walking, jogging, jumping, and sitting. However, whether the PPG sensor can be used for recognizing the finger-level gestures is still unknown.

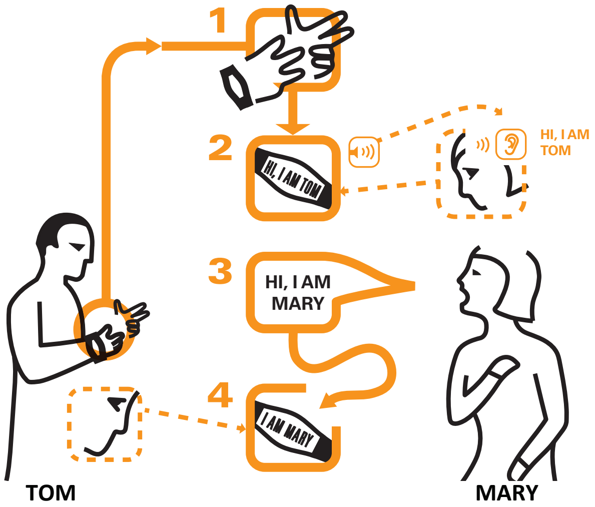


Fig. 1. Illustration of the automatic sign language translation using wearables in daily communications.

In this work, we are the first to demonstrate that low-cost PPG with the appropriately auxiliary help of the motion sensors in wearable devices could be exploited to accurately recognize sign language gestures, which are much more challenging than traditional gesture recognition due to involving subtle finger-level gestures. We study the unique PPG and motion sensor features resulted from finger-level gestures, and carefully devise a system that can effectively detect, segment, extract, and classify the sign language gestures based on the PPG measurements together with motion sensor measurements. The basic idea of our system is examining the blood flow changes resulted from finger-level gestures based on the PPG measurements, which are collected by low-cost PPG sensors available in wrist-worn wearable devices. As a comparison, we also investigate the limitation of only using motion sensors attached to the wrist area for finger-level gesture recognition. We show that the performance of the finger-level gesture recognition can have a significant improvement (i.e., around 10%) by combining the PPG and motion sensors.

The advantages of our approach are two-fold. First, our system could be easily applied to billions of existing wrist-worn wearable devices without extra cost, enabling every wrist-worn wearable device to recognize fine-grained gestures on users' fingers (e.g., sign language). Second, our system only relies on wrist-worn PPG and motion sensors, which directly obtain gesture related features without the impact of environmental changes (e.g., ambient light, sound, RF) and moderate body movements (e.g., walking, turning body, slow arm movements). Thus, it is more robust in practical scenarios. The main contributions of our work are summarized as follows:

- We demonstrate that PPG sensors in commodity wrist-worn wearable devices can be utilized to recognize fine-grained finger-level gestures. We develop the machine-learning approaches (i.e., GBT and ResNet) by leveraging the unique gesture-related PPG patterns captured by wearables on the wrist. Especially, transfer learning has been explored to significantly reduce the training efforts. We further show that motion sensors could be used as a complementary sensing modality to improve the gesture recognition accuracy. To our best knowledge, this is the first work recognizing finger-level gestures using commodity PPG sensors that are readily available in wrist-worn wearable devices.
- We explore the physical meaning and characteristics of PPG measurements collected from the PPG sensor on the wrist and develop a novel data extraction method that can precisely separate the PPG measurements caused by subtle finger movements from the continuous background noise caused by human pulses.
- We show that it is possible to accurately identify complicated finger-level gestures with minute differences (e.g., sign language gestures) by exploiting various types of features extracted from the unique gesture-related PPG patterns in different signal spaces (e.g., dynamic time warping, wavelet transform, Fourier transform).
- We reveal the limitation of using motion sensors for finger-level gesture recognition. We further develop a system that can adaptively integrate PPG and motion sensor data for finger-level gesture recognition based on different levels of body movements.
- We conduct experiments with 10 participants wearing our prototype consisting of two off-the-shelf PPG sensors, a motion sensor, and an Arduino board. We show that our system can achieve over 88% average accuracy of identifying 9 finger-level gestures from American Sign Language using only PPG sensor. In ideal scenarios without involving body movements, the average accuracy of our system could achieve 98% by integrating both PPG and motion sensor data. This suggests that our PPG-based finger-level gesture recognition system is promising to be one of the most critical components in sign language translation using wearables.

2 Related Work

In general, current techniques for gesture recognition can be broadly categorized into four categories (i.e., vision-based, RF-based, acoustic-based and body sensor based) as follows:

Vision Based. There are quite a few vision-based approaches have been developed to recognize hand/body gestures with the help of cameras. For example, Microsoft Kinect [3], [18] adopts the depth-sensor to measure the movements of the hand while performing hand gestures. LiSense [5] uses photodiodes on the floor to capture visible light changes and construct the user's 3D skeleton for gesture recognition. However, these approaches are sensitive to ambient light, and their accuracies are affected by the distance between the camera and the user's body. Leap motion [4], [19] utilizes the infrared LED cameras to capture the video of the hand, which can be translated into 3D points for gesture recognition without visible light. However, it still requires the user to use an additional device and line of sight to the user's gestures.

RF Based. RF-based approaches have become increasingly important due to the prevalent wireless environments and their device-free nature. Received signal strength indicator (RSSI) of WiFi has been utilized for gesture recognition since 2013. Wisee [9] builds the wireless prototype utilizing the Universal Software Radio Peripheral (USRP) and adopts the Doppler shifts of the wireless signals to achieve fine-grained gesture recognition. Wigest [20] uses WiFi RSSI's to detect human hand motions around a user device. Recently, channel state information (CSI) of WiFi has been widely studied for gesture recognition. WiDraw [8] harnesses the arriving angles of the WiFi signals received by the mobile device to track the user's hand trajectory. WiFinger [21] detects and identifies subtle movements of finger gestures by examining the unique patterns exhibited in CSI. However, these approaches either require dedicated and costly devices such as Universal Software Radio Peripheral (USRP) or can be easily affected by environmental changes such as people walking by.

Acoustic Based. Because most mobile devices have a strong capability of processing acoustic signals in nowadays, acoustic signals have been considered as an emerging sensing modality for gesture recognition. CAT [6] adopts a distributed Frequency Modulated Continuous Waveform (FMCW) that can accurately estimate the absolute distance between a transmitter and a receiver to continuously track gestures. Wang *et al.* [22] use the speakers and microphones of the mobile devices to perform the device-free tracking of a hand/finger based on the phase changes of the received acoustic signals. FingerIO [7] tracks the finger's dynamics by transforming the device into an active sonar system, which transmits inaudible Orthogonal Frequency Division Multiplexing (OFDM) signals and tracks the echoes of the finger using microphones. However, these approaches need to occupy the device's speaker/microphone or external audio hardware (e.g., nearby speakers), which is not always available in many real-world scenarios.

Body Sensor Based. In addition, several customized wrist-worn sensing platforms are designed to capture the hand gesture.

For example, Zhang *et al.* [13] proposes a framework utilizing the EMG sensor and accelerometer worn on the forearm to recognize Chinese sign language (CSL) gestures. Lu *et al.* [11] design a prototype utilizing the Surface Electromyography (SEMG) signals to control a mobile phone using predefined gestures. Zhang *et al.* [12] specially design a prototype that uses the Electrical Impedance Tomography (EIT) sensor equipped on either the wrist area or forearm area to recognize hand-level and finger-level gestures. However, these solutions need extra hardware support, and they are not compatible with existing mobile/wearable devices.

Another body of related work is using motion sensors in wrist-worn wearables to achieve hand and finger gesture recognition. For example, Xu *et al.* [14] leverage the accelerometer and gyroscope data from a wrist-worn device for recognizing arm-level, hand-level, and finger-level gestures. Wang *et al.* [15] examine the motion sensor data from a smartwatch to track the wrist micro-motions and infer what the user is typing on a regular keyboard. Wen *et al.* [23] design Serendipity that can distinguish five fine-motor gestures (e.g., pinching, tapping and rubbing fingers gestures) using the motion sensors in smartwatches. Gupta *et al.* [24] develop a method that can continuously recognize hand gestures using the motion sensors in a smart device despite the minor vibration from the user's hand. All these solutions only use motion sensors for gesture recognition. Therefore, they are sensitive to large body movements including forearm or body motions and cannot identify the fine-grained finger-level gestures, such as the sign language gestures with the existence of unexpected motion noises (e.g., body or arm movements).

PPG-based. Some works use the PPG signal to recognize human activities. For example, ActiPPG [16] can predict five types of human activities (i.e., standing, walking, jogging, jumping, and sitting) using raw PPG measurements. Biagetti *et al.* [17] also propose a real-time system for human activity recognition by using accelerometer and photoplethysmography (PPG) data. While these works show that PPG could be used for recognizing large body movements, whether it could be used to differentiate fine-grained finger-level gesture is unknown.

Different from previous work, we propose to innovatively use the photoplethysmogram (PPG) sensor, which is originally used for heart rate detection in most of the commodity wearable devices (e.g., smartwatch and wristband), to perform fine-grained finger-level gesture recognition and detection. To the best of our knowledge, it is the first wrist-worn PPG sensor based gesture recognition system. With the proposed scheme, we envision that most wearable device manufacturers would open the interface of PPG raw readings to developers soon.

3 Preliminaries & Feasibility Study

In this section, we discuss the preliminaries, design intuitions and feasibility studies of using PPG sensors in the wearable device for sign language gesture recognition.

3.1 Intuition of Finger-level Gesture Recognition Using PPG and Motion Sensors

3.1.1 Using PPG for Finger-level Gesture Recognition

During the past few years, more and more commodity wrist-worn wearables (e.g., smartwatches and activity trackers) are equipped with PPG sensors on their back. These wrist-worn PPG sensors are mainly designed to measure and record the user's heart rate. Specifically, a typical PPG sensor consists of a couple of LEDs and a photodiode/photodetector (PD), which detects the light reflected from the wrist tissue. The principle of PPG is the detection of blood volume changes in the microvascular bed of tissue. When light travels through biological tissue, different substances (e.g., skin, blood and blood vessel, tendon, and bone) have the different absorptivities of light. Usually, blood absorbs more light than the surrounding tissue. Therefore, by utilizing a PD to capture the intensity changes of the light reflected from the tissue, the wearable device can derive the blood flow changes in the wrist-area tissue and calculate the pulse rate or even blood pressure [25].

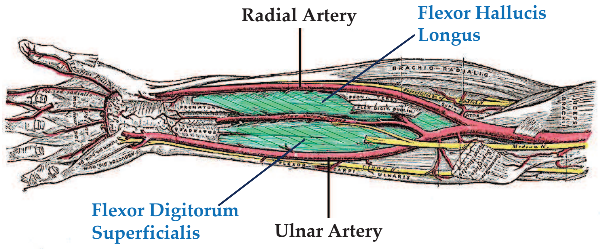
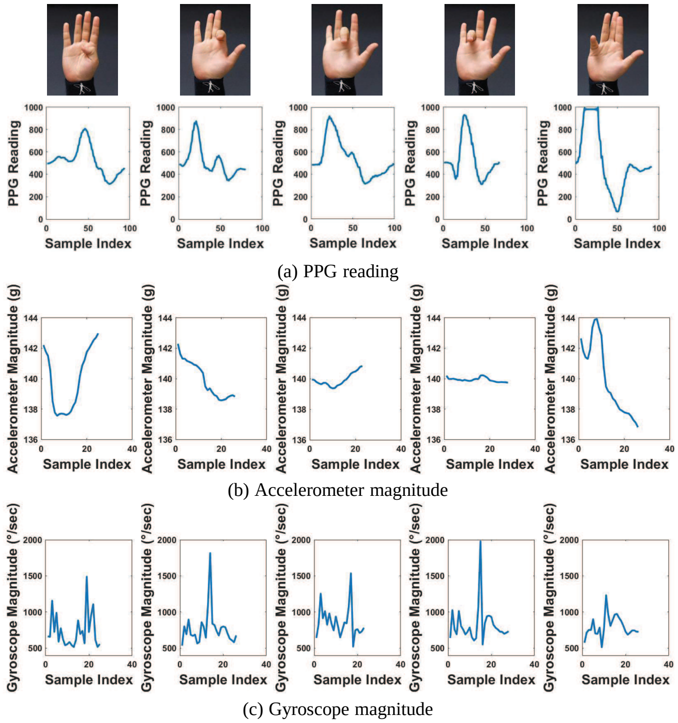


Fig. 2. Illustration of the finger movement related muscles in the anatomy of a human forearm.

The current use of PPG in wearables is limited to heart rate, pulse oximetry, and blood pressure monitoring. Such applications only focus on examining regular blood flow changing patterns in the radial artery and the ulnar artery and consider mechanical movement artifacts as noise [26]. In this work, we put forward an innovative idea of using readily available PPG in wearables for finger-level gesture recognition. We show that hand gestures, especially finger gestures (i.e., flexion, extension, abduction, and adduction), result in significant motion artifacts to PPG. The reason behind this is that the two major muscles controlling hand gestures [27], namely flexor digitorum superficialis and flexor hallucis longus, are right beside the radial artery and the ulnar artery as illustrated in Figure 2. Any hand or finger gestures would involve a series of complicated muscle and tendon movements that may compress the arterial geometry with different degrees. Since the blood absorbs most green lights, the changes of the light reflected from the wrist area present varying degrees of disturbances of the blood flow regarding the shapes and durations of PPG waveforms. Different from the existing work, we use the PPG data including not only motion artifacts, but also cardiac signals for the finger-level gesture recognition. We find that the finger-level gestures that we focus on have a similar spectrum as human cardiac movements. Therefore, filtering out cardiac signals would also remove subtle motion-related information in PPG measurements, which contain the distinct characteristics for differentiating finger-level gestures. In particular, the frequency-based filtering method in [16] removes the cardiac portion (0.5 $\sim$ 4 Hz), respiratory activity (0.2 $\sim$ 0.35 Hz) from the PPG measurements for recognizing the human activities such as walking (around 0.1 Hz). In our work, the finger-level gestures have the spectrum (0.5 $\sim$ 2 Hz). Therefore, we cannot use the traditional frequency-based filtering methods to isolate the motion artifacts from the cardiac signals. We note that the respiratory-related pattern has been removed using the band pass filter as mentioned in section 7 since its frequency is different from the cardiac signals and finger-level gesture.

It is important to note that most PPG sensors embedded in commodity wearable devices use green LEDs as the light source



(a) PPG reading

(b) Accelerometer magnitude

(c) Gyroscope magnitude

Fig. 3. Example of PPG, accelerometer magnitude and gyroscope magnitude associated with five finger-bending gestures in the feasibility study.

because they have much greater absorptivity for oxyhemoglobin and deoxyhemoglobin, compared to other light sources (e.g., red or infrared light) [26]. Current PPG sensors in off-the-shelf wearables are usually equipped with photodiodes to ensure accurate pulse estimation by increasing the diversity (i.e., monitoring blood flow changes at different locations on the wrist). Therefore, we use two green-LED PPG sensors in our prototype of wearable PPG sensing platform [28] to study and evaluate PPG based gesture recognition.

3.1.2 Improving Finger-level Gesture Recognition Using Wrist-worn Motion Sensors

Wrist-worn wearable devices such as fitness trackers and smartwatches are usually equipped with motion sensors (i.e., accelerometer and gyroscope) that are designed to capture the daily activities for extending user interfaces or infer users' motion states, including walking, running, driving, etc. In particular, the accelerometer measures accelerations of the wearable user's wrist and body movements, while the gyroscope provides angular velocities of the wrist rotations. Therefore, it is possible to distinguish a great number of wrist gestures by leveraging the accelerations and rotations obtained from wrist-worn motion sensors in wearable devices.

Hand gesture recognition using motion sensors in wearables have been extensively studied in recent years [14], [15], [23], [24]. It is natural to extend these technologies to facilitate finger-level gesture recognition. However, all the existing motion-sensor-based approaches are designed to distinguish wrist movements with significant displacements, which are not necessarily existing in finger-level gestures. Moreover, motion sensors in wrist-worn wearables are sensitive to motion noises (i.e., unintended body movements or forearm movements) while performing the finger-level gestures. Therefore, simply using motion sensors is not sufficient to distinguish the finger-level gestures. Considering that motion sensors can unveil more information of the finger-level gestures (such as vibrations from finger movements and forearm muscle movements), which are concealed from PPG measurements, we seek a solution leveraging motion sensors as a complementary measure to improve the performance of our PPG-based finger-level gesture recognition. In particular, our prototype system is implemented by adopting the 3-axis accelerometer and 3-axis gyroscope that are commonly found in most commodity wearable devices.

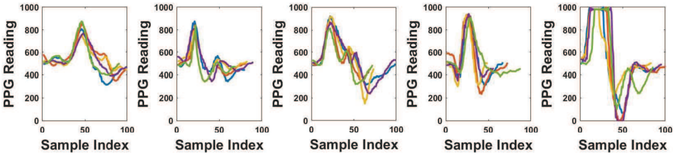


Fig. 4. Illustration of binding different fingers five times generates relatively similar PPG patterns,respectively.

3.2 Feasibility Study

PPG Sensor. In order to explore the feasibility of using PPG sensors in commodity wearables for finger-level gesture recognition, we conduct five sets of experiments on a sensing platform prototyped with two off-the-shelf PPG sensors (i.e., a photodiode sensor and a green LED) connecting to an Arduino UNO (Rev3) board [29], which continuously collects PPG readings at 100Hz and save them to a PC. During the experiments, a user wears a wristband to fix two off-the-shelf PPG sensors on the inner side of the wrist, and respectively bends each of his fingers as illustrated in Figure 3(a) to emulate the simplest elements of typical sign language gestures (e.g., number 1 to number 9). Specifically, in each set of the experiments, the user bends one of his finger 10 times with 8s between each bending. We record the process of the experiments using a video camera synchronized with the PPG measurements to determine the starting and ending time of each finger bending gesture.

We extract the PPG sensor readings within the time window between the starting and ending points identified in the video footage of each gesture and examine their changing patterns. As we expected, bending different fingers result in different unique patterns in PPG readings. Figure 3(a) presents an example of the unique patterns in PPG that correspond to bending and straightening different fingers, which is from one out of the two sensors. During our experiments, there's no intentional short pause between bending and straightening (i.e., they're performed in a consecutive way), which aligns with the normal performing style of the sign language. Moreover, as shown in Figure 4, we notice those same finger movements generate similar patterns, which demonstrates that it is possible to utilize readily available PPG sensors in wearables for fine-grained gesture recognition. We note that short pauses between bending and straightening may affect the gesture recognition performance if the pauses are not a part of the normal gesture. Since sign language users do not change their performing style often, our system is effective in general cases.

Motion Sensor. We next study the feasibility of using motion sensors for finger-level gesture recognition. Specifically, we examine the motion sensor (i.e., accelerometer and gyroscope) measurements by conducting the same experiments as those in the PPG sensor feasibility study. The motion sensors are installed on our prototype wrist-worn sensing platform, which collects data at 40Hz and saves them to a PC for further processing. We find

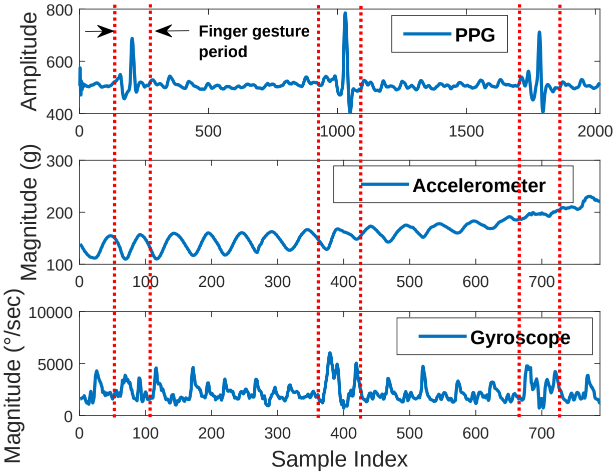


Fig. 5. Example of PPG reading associated with bending one specific finger (i.e., index finger) with moving forearm continuously.

that different bending-finger gestures can generate distinguishable magnitudes of the accelerometer and gyroscope readings as shown in Figure 3(b) and (c). We also notice that the same finger movement generates relatively similar patterns. Intuitively, the accelerometer can directly capture the three-dimensional acceleration resulted from the finger-level gestures which cannot be directly captured by the PPG sensor. Similarly, gyroscope could also provide three more-dimensional rotation information related to the finger-level gestures. In Section 8, we demonstrate that motion sensors can capture additional acceleration and rotation information of the finger-level gestures which can improve the overall performance.

Impact of Arm Movements. We further explore the feasibility of using PPG and motion sensor together for finger-level gesture recognition with the existence of the body movements. In particular, we conduct preliminary studies by asking the users to bend each finger successively while keep moving the forearm to emulate natural gestures, such as arm swinging when walking and arm lifting when checking time. We find that the forearm movement does not impact PPG readings, but the same forearm movement significantly affects the motion sensor readings and cause motion artifacts as shown in Figure 5. The insight is that the forearm movements involve little muscle movement in the wrist area, therefore there is little impact on the blood vessel and PPG readings, whereas the motion sensors always capture the motions irrelevant to finger-level gestures. The observation implies that PPG sensors are more robust to body movements in finger-level gesture recognition and the motion sensor can only be useful to finger-level gesture recognition in the ideal scenarios where little body movement is incurred.

4 Challenges & System Design

4.1 Challenges

In order to build a system that can recognize the sign language gestures using PPG and motion sensors in wearable devices, a number of challenges need to be addressed.

Re-using the PPG Sensors in Wearables for Sign Language Gesture Recognition. The PPG sensors in commodity wearable devices are specifically designed for monitoring pulse rate or blood pressure. The blood flow changes associated with the sign language gestures have a much shorter duration and do not have repetitive patterns compared to those caused by pulses. Our system thus needs to detect and discriminate the unique PPG patterns of different finger movements by re-using the low-cost PPG sensors in commodity wearable devices.

Gesture-related PPG Readings Interfered by Pulses. In this work, PPG readings corresponding to the sign language gestures are treated as target signals that our system wants to identify and examine. Therefore, the PPG readings resulted from pulses are considered to be the noise. Such noise always exists and sometimes has intensity comparable to that of the signals caused by the sign language gestures. Our approach should be intelligent enough to separate relevant useful signals from the complicated noise caused by pulses.

Robust Sign Language Gesture Recognition Using Limited Sensing Modality on the Wrist. It is also challenging to achieve high accuracy in sign language gesture recognition by using the readily available but coarse-grained sensing modalities (i.e., PPG and motion sensors). Commodity wearable devices usually have a limited number of PPG sensors that are placed very close to each other. Such layout limits the coverage of the PPG sensors on the wrist and the diversity of sensor readings, which could significantly impact the performance of gesture recognition. In addition, motion sensors in wearables can only benefit the gesture recognition performance when there is no significant body movement. Thus, we need to explore the critical features in PPG and motion sensor readings in various domains to achieve reliable sign language gesture recognition.

Reducing Training Effort for Practical Usage. Training effort can reflect the ease of use of the system. Long-time and tedious training procedure can significantly impact the user experience. Our system takes this into consideration and adopts advanced machine-learning approach to provide robust and accurate sign language gesture recognition with the requirement of just a few training data from users, which is critical to real-use scenarios.

4.2 System Overview

The basic idea of our system is examining the blood flow changes captured by readily available PPG sensors in commodity wrist-worn wearable devices to differentiate the sign language gestures. In addition, our system takes the acceleration and rotation measurements captured by motion sensors as the opportunistic measure to improve the accuracy of gesture recognition when there is no body movement interfering the motion sensor readings. As illustrated in Figure 6, our system first takes as inputs the PPG, acceleration, and rotation measurements from wrist-worn PPG and motion sensors, respectively. Next, our system conducts the *Sensor Selection* to allow the system to integrate the motion sensor readings dynamically, depending on whether there are body movements detected based on the magnitude of the gyroscope measurements. When body movements are detected, our system only selects the PPG sensor for gesture recognition to avoid large errors; otherwise our system selects both motion and PPG sensors to achieve better performance. After determining the right sensor to use, the *Coarse-grained Gesture Detection and Reference Sensor Determination* module is performed to determine whether there is any gesture being performed based on the PPG signal energy. Meanwhile, the system automatically determines the *Reference Sensor*, which is the PPG sensor presenting the most significant (i.e. containing

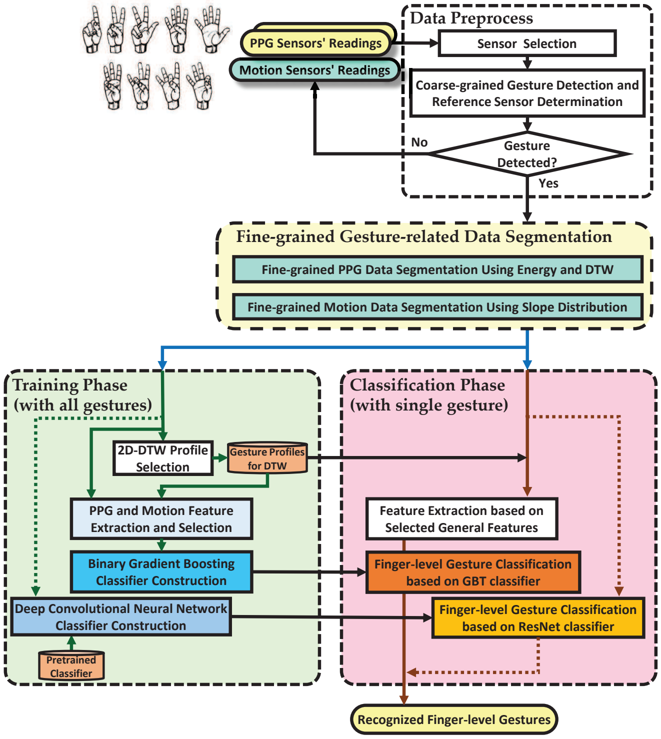

PPG Sensors' Readings
Motion Sensors' Readings
Data Preprocess
Sensor Selection
Coarse-grained Gesture Detection and Reference Sensor Determination
No
Gesture Detected?
Yes
Fine-grained Gesture-related Data Segmentation
Fine-grained PPG Data Segmentation Using Energy and DTW
Fine-grained Motion Data Segmentation Using Slope Distribution
Training Phase (with all gestures)
Classification Phase (with single gesture)
2D-DTW Profile Selection
Gesture Profiles for DTW
PPG and Motion Feature Extraction and Selection
Feature Extraction based on Selected General Features
Binary Gradient Boosting Classifier Construction
Finger-level Gesture Classification based on GBT classifier
Deep Convolutional Neural Network Classifier Construction
Finger-level Gesture Classification based on ResNet classifier
Pretrained Classifier
Recognized Finger-level Gestures

Fig. 6. Overview of our sign language gesture recognition system.

more energy) gesture-related signal patterns compared to those related to pulses.

For each coarse-grained gesture segment, the system uses *Fine-grained Gesture-related Data Segmentation* to extract the gesture-related signal patterns from both PPG and motion sensors, respectively. Specifically, we perform the fine-grained PPG data segmentation based on the short-time energy and Dynamic Time Warping (DTW) distance to accurately extract the PPG data segments of gesture-related patterns in *Fine-grained PPG Data Segmentation Using Energy and DTW* module. If motion sensor readings are involved, the system further performs the fine-grained motion sensor data segmentation based on Kullback-Leibler divergence of the signal slope distributions to accurately extract the motion sensor data segments containing gesture-related patterns in this module. After the fine-grained data segmentation, we have developed two classifiers (i.e., GBT and ResNet) to deal with the hardware computational capability limitation. In Section 8, we show that both classifiers have a similar performance. Compared to the GBT classifier, the ResNet classifier is an alternative solution for the devices having the limited computational capability to extracting features in real-time since ResNet does not require the process of feature extraction. Then, the data processing of our system is separated into two phases: *Training Phase* and *Classification Phase*.

Training Phase. In this phase, we collect labeled PPG and motion sensor measurements for each gesture of a user and build the binary gradient boosting tree (GBT) classifier for each user. Specifically, after segmentation, our system calculates the 2D-DTW distances between every two PPG segments for every gesture in the *2D-DTW Profile Selection* and selects three profile PPG segments that are most representative for each gesture (i.e., having the minimum average 2D-DTW distance to other segments of the same gesture). The selected profile PPG segments will be used to calculate the DTW features in the *Classification Phase*. Meanwhile, in the *PPG and Motion Feature Extraction and Selection*, the system performs the PPG and motion sensor feature extraction and selection to derive a variety of features in different signal spaces (e.g., discrete wavelet transform, fast Fourier transform). After that, the system selects the critical features that can effectively capture the unique gesture-related PPG and motion patterns for each gesture. Because the selected critical feature sets are optimized for each gesture, the system further derives a superset of the selected critical features (i.e., general features) to ensure the system performance. Next, we perform the *Binary Gradient Boosting Classifier Construction* to train a binary classifier of each user for each target gesture using gradient boosting. In addition, we develop a multivariate deep CNN (ResNet) classifier for each user for sign language gesture recognition. Specifically, our system combines the segmented measurements directly as the multivariate input and perform the *Deep Convolutional Neural Network Classifier Construction* to train the ResNet classifier for each people. To explore the possibility of using the participant-independent model, we adopt the transfer learning technique by integrating the *Pre-trained Classifier* of a user while training, which significantly reduces the training efforts for other users.

Classification Phase. In the *Classification Phase*, our system collects the sensor measurements in real time and determines which target sign language gesture has been performed based on the classification results. Specifically, when our system adopts the GBT classifier, it extracts the selected general features from the selected sensor data segments of the current user and performs the classification using *Finger-level Gesture Classification Using GBT* by using the binary gradient boosting classifiers of the current user generated in the training phase for all the gestures in parallel. Each binary classifier generates a confidence score, and the system takes the target gesture having the highest confidence score as the recognized gesture. When our system adopts the ResNet classifier, it performs the *Finger-level Gesture Classification Using ResNet*, which directly uses the segmented sensor measurements of the current user as the multivariate inputs for his/her classifier to determine which target gesture has been performed.

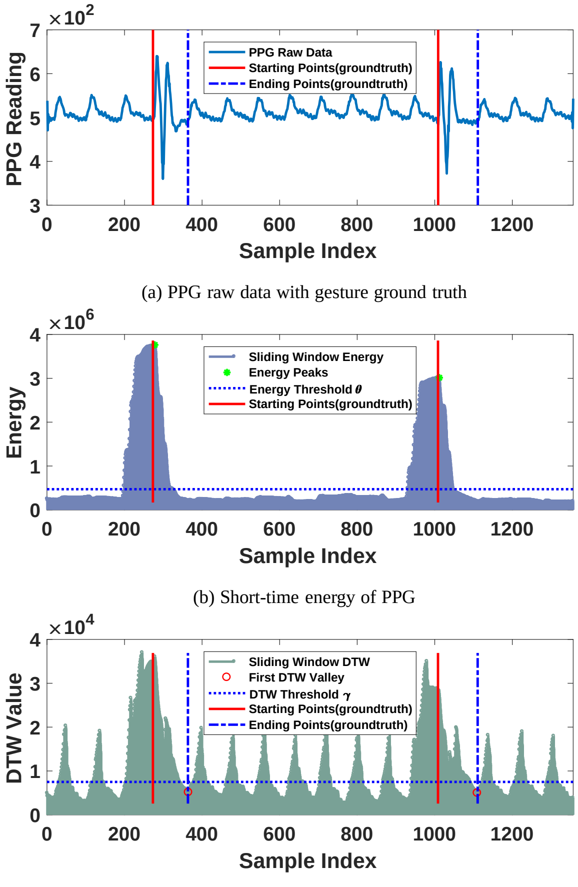
5 Fine-grained Data Segmentation

Accurate sign language gesture recognition requires to pinpoint the starting and ending points of the gesture from the related sensor measurements. In this section, we discuss how to achieve fine-grained data segmentation based on the raw PPG and motion sensor data segments that have been verified to contain significant gesture-related patterns through the *Data Preprocess* discussed in Section 7.

5.1 Fine-grained PPG Sensor Data Segmentation

5.1.1 Starting Point Detection Using Energy

We first determine the starting time of the gesture. Due to the consistent existence of pulse signals in PPG measurements, it is difficult to remove the pulse signals without jeopardizing the details of the gesture-related readings, which are critical to characterizing the starting and ending points of a specific gesture. In order to accurately determine the starting point, we seek an effective detection approach to mitigate the impact of pulse signals. We find that the gesture-related PPG signals are usually stronger than those caused by pulses as illustrated in



(a) PPG raw data with gesture ground truth

(b) Short-time energy of PPG

(c) Short-time DTW distance to pulse profile

Fig. 7. Example of detecting starting and ending point of a gesture-related PPG measurements using energy and DTW.

Figure 7(a), because gestures usually involve dynamics of major forearm muscles/tendons close to the sensor on the wrist. Inspired by the above observation, we design an energy-based starting point detection scheme to effectively estimate the starting of gesture-related PPG signals without removing the interference of pulses.

The basic idea of our energy-based starting point detection method is to determine the time corresponding to the local maximum of the short-time energy of PPG signals. The reason behind this is that when using a sliding window with the same length of a signal to calculate the short-time energy of the signal, the energy reaches its maximum value when the signal entirely falls into the window. Therefore, by carefully choosing the size of the sliding window (e.g., the average length of target gesture-related signals), the starting point of the gesture-related signals would be the same time when the short-time energy of the signals reaches its maximum. In particular, given the data segment containing gesture-related PPG signals $P(t)$ from the *Coarse-grained Gesture Segmentation* (Section 7), the starting point detection problem can be formulated as the following objective function:

$$\arg\max_{\tau} \; (P(\tau)-\mathbf{1}\theta)P(\tau)^T, \qquad (1)$$

where $P(\tau)=[p(\tau),p(\tau+\delta)\cdots,p(\tau+W)]$, $p(\tau)$ denotes the amplitude of the PPG signal at time τ, δ represents the PPG sensor sampling interval, W is the length of the sliding window, θ is the threshold used to avoid finding the local maximum energy resulted from pulse signals, $\mathbf{1}$ is an all-one vector of the same length as $P(\tau)$, and T indicates the transpose operation. The above problem can be easily solved through simple 1-D searching within the period derived from coarse-grained gesture segmentation.

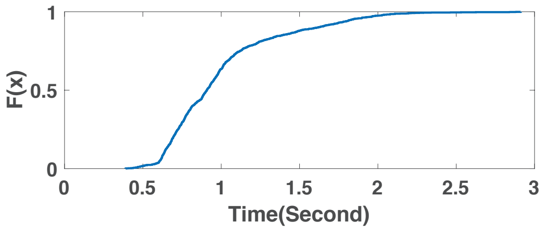


Fig. 8. Preliminary study: CDF of the duration of 1080 gestures from 3 users.

Through our preliminary study on the time length of 1080 sign language gestures performed by three users as shown in Figure 8, we find that the length of gesture-related signals has the range between 0.7s and 1.4s with an average of 1.2s. Therefore, we empirically determine the length of the sliding window as 1.2s to ensure the accuracy of our energy-based starting point detection. Note that the threshold θ is user-specific and needs to be dynamically determined by the maximum short-time energy of the PPG signals when there is no gesture detected in the *Coarse-grained Gesture Detection*. Figure 7(b) illustrates the short-time energy corresponding to the PPG signals in Figure 7(a). We can clearly see that the energy peaks in Figure 7(b) are very close to the ground truth observed from the synchronized video footage, suggesting that our algorithm could promisingly capture the starting point of gestures in the PPG measurements.

5.1.2 Ending Point Detection Using DTW

It is more challenging to detect the ending point of a gesture in the PPG signal than the starting point because the muscles are more relaxed at the end of the gesture and the corresponding PPG signals are usually weaker than those at the beginning of the gesture. As illustrated in Figure 7(a), the PPG measurements around the ending point do not have significant patterns that can facilitate the ending point detection. However, we find that gesture-related PPG signals are usually immediately followed by pulse signals, which are very clear and easy to identify. Hence, instead of directly locating the ending point based on PPG readings, we design a DTW-based ending point detection scheme, which aims to identify the starting time of the first pulse signal following the gesture-related signal. We employ the dynamic time warping (DTW) to measure the similarity between the user's pulse profile P_{pulse} and the PPG measurements collected after the already-detected starting point of the gesture. Intuitively, the time when the DTW value reaches the minimum is the starting time of the pulse signals and also the ending point of the gesture-related signals. We adopt DTW because it can stretch and compress parts of PPG measurements to accommodate the small variations in the pulse signals. To summarize, this ending point detection problem is defined as follows:

$$\arg\min_{t} \; DTW(P(t),P_{pulse}), \quad s.t., \tau < t \le \tau+W_p, \qquad (2)$$

where $DTW(\cdot,\cdot)$ is the function to calculate the DTW distance, $P(t)$ has the same definition as $P(\tau)$ in Equation 1, W_p is the time duration for the gesture, and τ is the detected starting point. After searching the DTW distances for all $P(t)$, we find the time index of the first local minimum in the DTW distances (i.e., the starting time of the first pulse after the gesture) as the ending point of the gesture-related signals. Figure 7(c) presents the DTW between a selected pulse profile and the raw PPG measurements in

Figure 7(a) with $W_p = 0.88$s. From the figure, we can observe that the time indices of the detected first local minimum DTW values are very close to the ground truth of the ending time of the two gestures, which demonstrates the effectiveness of the DTW-based ending point detection scheme.

Extracting Pulse Profiles. The pulse profile P_{pulse} can be extracted from the PPG measurements that are collected when there is no gesture performed (e.g., at the beginning of the training phase). In particular, we first detect the pulse signal peaks in the PPG measurements. Given the fact that a typical PPG pulse signal always has a peak, if the pulse signal peak is located at t_p, so the PPG measurements between $[t_p - t_d, t_p + t_s]$ are identified as the user's pulse profile. In this work, we respectively choose $t_d = 0.2$s and $t_s = 0.6$s based on the duration of diastole (i.e., 0.15s$\sim$ 0.26s) and systole (i.e., 0.44s$\sim$ 0.74) phases of the vascular system reflected in a typical PPG pulse signal [30], which can effectively extract all users' pulse profile.

5.2 Segmentation on Inconspicuous Gesture-related Patterns

Our DTW-based ending point detection can accurately determine the ending point if the gesture-related PPG pattern has significant amplitudes compared to those of the pulse-related patterns. However, in rare cases, the gesture-related PPG patterns may not have significant amplitudes when the sensor is at the locations far away from the arteries. Note that such inconspicuous patterns are not easy to be extracted as their boundaries with pulse-related patterns are very vague, but they still contain rich information that could greatly facilitate gesture recognition. In this work, we find that when using two PPG sensors close to each other on the wrist, at least one of the sensors can generate gesture-related PPG patterns with significant amplitudes. Inspired by this observation, we adopt a reference-based approach to accurately determine the ending point for the inconspicuous gesture-related PPG patterns.

In particular, assuming our system identifies the ending point t_R on the sensor R with significant gesture-related PPG patterns (i.e., *Reference Sensor* discussed in Section 7) using our DTW-based method, the system further derives the ending point at the other sensor D as $t_D = t_R + \Delta T$, where ΔT is the time delay between the ending points on sensor R and sensor D. According to our empirical study, ΔT is nonzero and stable between two sensors across different gestures. The insight is since muscles and tendons at different locations of the forearm compress the arteries with different pressures and durations when performing a gesture, the gesture-related patterns captured by the PPG sensors at different locations will last different time periods. Because the system can always find multiple gestures that generate significant PPG patterns on both sensors, ΔT can be easily estimated in the *Training Phase* by calculating the average time difference of the ending points from the gestures where both sensors are determined to be *Reference Sensors*.

5.3 Fine-grained Motion Sensor Data Segmentation

Next, we perform the data segmentation on motion sensor data to identify fine-grained starting and ending points of a sign language gesture. Intuitively, the gestures involving the movements of the major forearm muscles/tendons on the wrist induce strong fluctuations in motion sensor readings, which usually have relatively low and stable amplitudes when there is no gesture performed. In our experiments, we observe that the amplitude of motion sensor readings sharply increases at the starting point of a gesture, then keeps fluctuating during the gesture, and finally decreases sharply to a low and stable level at the ending point of the gesture. Both starting point and ending point are obtained through *Coarse-grained Gesture Detection and Segmentation* in Section 7. These observations indicate that in one coarse-grained segment, the distributions of motion sensor readings before and after the starting/ending point of a gesture have the most significant difference, respectively. Therefore, we design a Kullback-Leibler (K-L) divergence technique-based detection scheme to effectively estimate the starting and ending point of a gesture from motion sensor signals in the coarse-grained segments. The basic idea is that given two sliding windows with the same length traversing the coarse-grained segment, the starting and ending points of a gesture are determined as the time points when the difference of the distributions derived from these two windows present two significant peaks in tandem, respectively.

In particular, we respectively calculate the distributions of quantized motion sensor reading slopes in the sliding windows before and after each time point t_j, which can be denoted as $B(K_{j-1})$ and $A(K_j)$, $j = [1, ..., J]$, where J is number of time points in the coarse-grained segment. Then, we calculate the difference between the slope distribution $B(K_{j-1})$ and $A(K_j)$ of each time point t_j using K-L divergence. The K-L divergence between these two distributions are derived as: $D_{KL}(B(K_{j-1})|A(K_j)) = \sum_{q \in Q} B(K_{j-1} = q) \ln \frac{B(K_{j-1}=q)}{A(_{k_j=q})}$, where Q is the set of all possible values for quantized motion sensor reading slopes. The insight is that the motion sensor readings have a sharp change around the starting and ending points. So, the local maximum slope distribution difference determined by K-L divergence corresponds to the starting and ending points. It's important to note that $1/16$ window overlapped is used in our algorithm for optimizing the performance.

6 Sign Language Gesture Classification

In this section, we introduce how to extract the PPG and motion sensor features that can facilitate sign language gesture recognition using our Gradient Boosting classifier. In addition, we explore the advantage of using the Deep Residual Network for sign language gesture recognition, which could leverage the transfer learning technique to reduce users' training effort significantly.

6.1 PPG and Motion Feature Extraction

PPG Feature Extraction. To capture the characteristics of unique gesture-related PPG patterns, we explore the efficacy of different kinds of features including typical temporal statistics (e.g., mean, variance, standard deviation (STD)), cross-correlation, autoregressive (AR), dynamic time warping (DTW), fast Fourier transform (FFT), discrete wavelet transform (DWT), and Wigner Ville distribution as listed in Table 1. The features can be categorized into three types: *Time Domain*, *Frequency Domain*, and *Time-Frequency Domain*, which are designed to capture the detailed characteristics of the gesture-related PPG patterns across different frequency and time resolutions. While the *AR Coefficients*, *FFT*, *DWT*, *WVD*, and most of the *Classic Statistics* are all focusing on analyzing an individual sensor's measurements, the *Cross Correlation* and *2D-DTW* are promising for characterizing the unique gesture-based PPG patterns in terms of the relationship between a pair of sensors. Moreover, our *Time-Frequency (TF) Domain*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2962760, IEEE Transactions on Mobile Computing

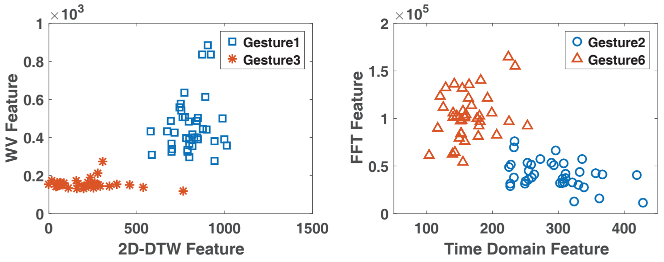
TABLE 1
List of Extracted PPG Features.

Category	Features (# of features)	Description
Time Domain	**Classic Statistics** (4): mean, peak-to-peak, RMS, variance	Descriptive statistics of each segment, reflecting the statistical characteristics of the unique gesture-related patterns.
	Cross Correlation between Sensors (9)	A vector of cross correlation coefficients between the segments from two PPG sensors based on a sequence of the lag values, characterizing the relationship between two PPG sensors in a gesture.
	2D-DTW to Gesture Profiles (9)	Similarity between PPG measurements from two sensors (i.e., 2D) and the corresponding gesture profiles, directly capturing the temporal shape characteristics of the unique gesture-related patterns.
Frequency Domain	**Fast Fourier Transform (<5Hz) (6)**: skewness, kurtosis, mean, median, var, peak-to-peak	Statistics of frequency components in the specific low frequency band, analyzing the unique PPG patterns in frequency domain.
Time-frequency Domain	**Discrete Wavelet Transform (4)**: mean, peak-to-peak, RMS, variance	Statistics of the third level decomposition of the wavelet transform using the Harr wavelet, revealing the details of gesture-related patterns at interested time and frequency scale.
	Wigner Ville Distribution [31] (13): first-order derivative, frequency and time when the signal reaches the maximum, maximum energy (E^i_{max}) / minimum energy (E^i_{min}), differential energy ($E^i_{max} - E^i_{min}$), STD^i and AV^i of the energy within the i^{th} sliding window	Fine-grained time-frequency features with high resolutions, capturing details of gesture-related patterns having short time duration.
	Autoregressive Coefficients [32] (9)	Time variant coefficients that can capture the characteristics of gesture-related patterns independent of the patterns' time scales.

features include three major TF types (i.e., non-parametric linear TF analysis (DWT), non-parametric quadratic TF analysis (WVD), and parametric time-varying based metric (AR)), which can well capture the dynamics of gestures in PPG measurements. In total, we extract 54 different features from each PPG sensor. Note that in order to calculate the 2D-DTW feature, our system first performs *2D-DTW Profile Selection* in the *Training Phase*, which calculates the 2D-DTW distance between every two segments for every gesture in the training data and selects three segments that have the minimum average 2D-DTW distance to other segments of the same gesture as the profile for later use in the *Classification Phase*.

Motion Feature Extraction. The time domain features such as the Mean, Max, Min value, Variance of each segment in the motion sensor readings have been demonstrated to be able to effectively capture the distinguishable signal patterns from different people, who perform the same hand gestures as shown in the research [33]. To further characterize the unique gesture-related acceleration and rotation patterns, we also explore the efficacy of other features including typical temporal statistics (e.g., peak2peak, root mean square (rms)) in the time domain, which could reveal the detailed characteristics of the gesture-related acceleration and rotation measurements. Since those features can effectively capture the geometrical characteristics of the gesture-related signal segments from the motion sensor, therefore we adopt those features as the motion features for the gesture classification.

Feature Selection. Our system further employs the elastic net feature selection method [34] in the *Training Phase* to automatically choose the most discriminative ones from our extracted features. In particular, the system respectively performs the elastic net feature selection on the PPG and motion sensor features corresponding to every target gesture. Based on the one-stand-deviation rule [35], our system keeps the most significant highly correlated features and eliminates noisy and redundant features to shrink the feature set and avoid overfitting. Next, in order to generalize the features set for classifying all target gestures, our system integrates the features selected for each target gesture and generates a general feature set $\mathbb{F}$ as follows: $\mathbb{F} = F(g_1) \cup \ldots \cup F(g_{n)}$, where $F(g_n)$ is the selected feature set of the n_{th} target gesture g_n. After the feature selection and integration, we have 66 *Determined General Features* (i.e., 54 PPG features and 12 motion sensor features) in $\mathbb{F}$, which will be used in the *Classification Phase*. Figure 9 illustrates that our features can effectively capture different characteristics of PPG and motion sensor patterns for distinguishing different gestures.



(a) Gesture 1 & gesture 3

(b) Gesture 2 & gesture 6

Fig. 9. Example of different sign language gestures and corresponding features.

6.2 Finger-level Gesture Classifier Using Gradient Boosting Tree

Next, we build a binary classifier for each target gesture by using the Gradient Boosting Tree (GBT) for each user. We choose GBT mainly because 1) GBT is famous for its robustness to various types of features with different scales, which is the exact case in our project (e.g., the mean value of the PPG signal reading of the gesture period is around 500, and the autoregressive coefficients are the numbers fluctuated around 0 with value less than 1). 2) GBT classifier is robust to the collinearity of feature data. Because our features are heterogeneous across different domains, it may result in unexpected correlation or unbalance ranges that possess the collinearity. Therefore, GBT would eliminate the efforts to normalize or whiten the feature data before classification [36].

Given N training samples $\{(x_i, y_i)\}$, where x_i and y_i represent the gesture-related feature set and corresponding label with respect to one specific gesture (i.e., y_i = 1 or -1 represents whether x_i is from this gesture), GBT seeks a function $\phi(x_i) = \sum_{m=1}^{M} \omega_m h_m(x_i)$

to iteratively select weak learners $h_j(\cdot)$ and their weights ω_j to minimize a loss function as follows: $\mathbf{L} = \sum_{i=1}^{N} L(y_i, \phi x_i)$. Specifically, we adopt the GBT implementation from the library of SQBlib [37] for gesture-related feature training. Specifically, the loss function $L(\cdot)$ is chosen as the exponential loss $L = e^{-y_i \phi x_i}$ that applies enough shrinkage (i.e., 0.1) and number of iterations (i.e., $M = 2000$), and the sub-sampling of the training dataset is a fraction of 0.5. The above parameters adopted in GBT are optimized in terms of the speed and accuracy based on our empirical study. Once the loss function is determined, we next will build a binary gradient classifier $b_k(\cdots)$ for each profiled gesture $g_k, k = 1, \cdots, K$ to complete the *Training Phase*, and each binary gradient classifier will output a score for the testing feature set. The reason of using binary classifier is that binary classifier has high accuracy with distinguishing one gesture from other gestures, whereas a multi-classifier has relative lower accuracy when performs the same classification task [38].

In *Classification Phase*, our system uses the binary classifiers of the current user for all the gestures in parallel to classify previously unseen gesture-related feature set x. Specifically, we will obtain different scores returned from each binary classifier, and choose the label k of binary classifier $b_k(x)$ with highest score as the final classification.

6.3 Finger-level Gesture Classifier Using Deep Residual Network (ResNet)

Recent years have witnessed the success of Deep Neural Networks (DNNs) on Time Series Classification (TSC). In this work, we especially adopt the deep Residual Network (ResNet) architecture proposed in [39], which provides excellent performance in both univariate and multivariate TSC [40]. In particular, ResNet is a variant of convolution neural network (CNN) with a linear shortcut to link each residual block to its input, which reduces the vanishing gradient effect and thereby eases the DNN training procedure [41].

Particularly, we design a ResNet consists of 3 residual blocks followed by a Global Average Pooling (GAP) layer and a softmax layer to classify the output of GAP layer (11 layers in total). Each residual block has 3 convolution layers, and the output of each convolution layer then undergoes batch normalization and ReLU activation before feeding into the next layer. The three convolution layers are set to include 64 filters with the lengths of 8, 5 and 3, respectively. Moreover, the output of each residual block is added to its input as the updated input to next residual block, and the last residual block of each convolution layer connects to GAP layer that averages the time series over whole time dimension and helps minimize overfitting. The adaptation of GAP instead of the traditional fully connected (FC) layer not only significantly reduces the total number of parameters in the network model but also enables the class activation map (CAM) to identify the most significant part of the time series input for classification.

In the *Training Phase*, we train the ResNet for each user and feed the ResNet with a user's input data of N samples $\{(TS_i, y_i)\}$, where TS_i and y_i represent the time series input data (i.e., both raw PPG and motion sensor segments) and the corresponding label with respect to one specific gesture (i.e., y_i = k represents that y_i is from the gesture $k, k = 1, \cdots, K$). Particularly, we use the shuffle split cross-validation method to randomly select the specified number of the training data from the input dataset of the current user and use the rest of the user's data as the testing data. In the meantime, the classifier is run by the testing data simultaneously. After each epoch of the training phase, we track the accuracy with the currently trained model applied on the testing data, and the model with the highest accuracy will be chosen as the final model after all the epochs. In our implementation, we adopt 16 as the batch size and 1500 as epoch.

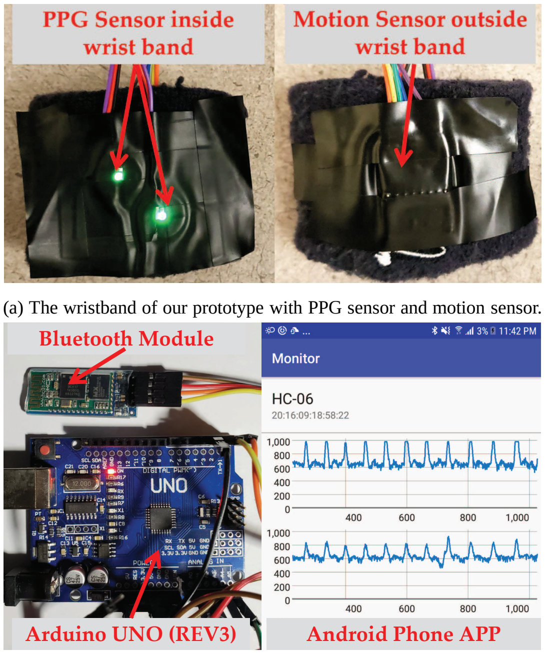
Reducing Training Effort Using Transfer Learning. Our work also explores transfer learning across different users to reduce the training effort in terms of the size of the training dataset. In particular, we fix the architecture of the ResNet with the invariant number of neurons in each layer (except the softmax layer) and retain the weights of the higher layer neurons since the lower layers in the ResNet refer to general features, while higher layers captures more dataset-specific (i.e., the specific people) features. With this fixed architecture, the pre-trained model on one user's dataset (i.e., source dataset) could achieve the similar performance on another user's dataset (i.e., target dataset) with much smaller training dataset size, and the retraining process only needs to fine-tune on the values of the parameter without modifying the structure of the hidden layers in ResNet. In particular, we retrain the classifier for the new people from the third residual block to GAP layer to softmax layer in the ResNet. In testing phase, we apply the retrained model of the new user on his/her testing dataset to get the classification results.

7 Data Preprocessing

In this section, we present **three** components that are critical to **sensor selection** and fine-grained data segmentation.

Sensor Selection. Body movements (e.g., forearm swing and body rotation) may induce irrelevant motion artifacts in the motion sensor measurements, which significantly impacts the performance of sign language gesture recognition. However, such body movements have little impact on the PPG measurement as aforementioned *Feasibility Study*. Hence, it is critical to determine the right moment to integrate PPG measurements with motion sensor measurements for sign language gesture recognition in our system. Our extensive experimental study finds that body movements usually introduce significant fluctuation on both accelerometer and gyroscope measurements simultaneously, whereas finger-level gestures alone do not. Therefore, we employ a threshold-based approach to determine the presence of body motion. In particular, we calculate the magnitude of the gyroscope measurements. If it's below an amplitude threshold, θ_a, before the starting point of a sign language gesture, our system takes both PPG and motion sensor measurements for data segmentation and gesture recognition. Otherwise, only PPG measurements are used. Note that the threshold is mainly determined by the sensitivity of the motion sensor. In this paper, we empirically determine $\theta_a = 38.4$ radians per second based on our experimental study with 10 participants. Our evaluation results (Section 8.3) demonstrate that this threshold can help to render good gesture recognition accuracy.

Coarse-grained Gesture Detection and Segmentation. To facilitate the fine-grained data extraction, our system pre-processes the raw PPG measurements to 1) determine whether there is a gesture performed or not based on the short-time energy of the PPG measurements; 2) and extract the PPG and motion measurements that surely include the whole gesture-related patterns. Specifically, the system first applies a high-pass filter to the raw PPG measurements to mitigate the interference of pulses. The reason to use the high-pass filter is that the finger-level gestures have more high-frequency components compared to the pulses, which are



(a) The wristband of our prototype with PPG sensor and motion sensor.

(b) Arduino board with the Bluetooth and our Android smartphone APP

Fig. 10. Prototype wrist-worn PPG sensing platform.

usually under 2Hz [42]. In this work, we build a Butterworth high-pass filter with the cut-off frequency at 2Hz. Note that we only use the filtering technique in the coarse-grained gesture detection, because the filter removes the low-frequency components of both pulse and gesture-related signals, which negatively impact the gesture recognition accuracy. Then the system decides whether there is a gesture performed or not depending on if the short-time energy of the filtered PPG measurements crosses a threshold τ or not. We set the threshold to $\tau = \mu + 3\delta$, where μ and δ are the mean and standard deviation of the short-time energy of the filtered PPG measurements collected during the time when the user is asked to be static (i.e., at the beginning of the training phase). When the system detects a gesture at t_g, we employ a fixed time window W_c to extract the raw PPG and motion sensor measurements within $[t_g, t_g + W_c]$ for the fine-grained segmentation. We set $W_c = 4.5$s to ensure the window can cover all possible duration of gestures that we have observed in our preliminary study as shown in Figure 8. **Reference Sensor Determination.** Intuitively, significant gesture-related PPG patterns could result in accurate data segmentation. However, we notice that the intensity of gesture-related PPG patterns is sensitive to the locations of sensors on the wrist, which may not be significant enough for segmentation. The insight is that the PPG sensors can capture more significant changes of reflected light when they are closer to the arteries that are directly compressed by muscles and tendons. Through our extensive tests, we find that two PPG sensors at a close distance on the wrist can already provide good diversity, and at least one of them can provide gesture-related PPG signals that have the stronger intensity than that of pulse-related signals. Therefore, in this work, we employ a two-sensor approach and determine which sensor could be the *Reference Sensor* having the significant gesture-related PPG patterns, which will be taken as the input for the fine-grained data segmentation. Specifically, we examine the short-time energy of the extracted PPG measurements and determine whether a sensor is a *Reference Sensor* or not depending on if its short-time energy exceeds the threshold θ, which has been defined in Equation 1.

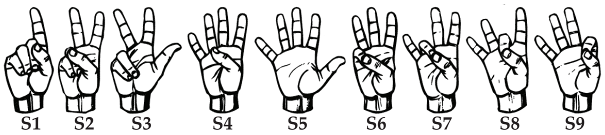


Fig. 11. American Sign Language of number one to nine.

8 Experiment and Evaluation

8.1 Experimental Methodology

8.1.1 *Wearable Prototype*

We notice that existing manufacturers do not provide direct access to raw PPG readings; instead, they only provide the computed heart rate. Therefore, we design a wearable prototype to mimic the layout of PPG sensors in commodity wearables to demonstrate that our system can be applied to the existing wearable products without extra efforts. Our prototype has two commodity PPG sensors (with single green LED) and a motion sensor (i.e., MPU-6050 with a three-axis gyroscope and a three-axis accelerometer) connected to an Arduino UNO (REV3) board, which exchanges data with our android app through Bluetooth as shown in Figure 10. The two PPG sensors are placed closely to each other and fixed to the inner side of a wristband, while the motion sensor is placed outside of the wristband. The training phase is done offline using MATLAB. The trained GBT and ResNet classifiers are deployed on our app to perform the testing phase of our sign language gesture recognition. In the experiments, we adopt various sampling rates (i.e., 30Hz, 40Hz, 60Hz, 80Hz, and 100Hz) to evaluate the system. Unless mentioned otherwise, the default sampling rate is set to 100Hz.

8.1.2 *Data Collection*

We recruit 10 participants (7 males and 3 females with the age range from 20 to 40) to perform the sign language gestures for evaluation. To better demonstrate the effectiveness of recognizing finger-level gesture, we focus on the elementary gestures from American Sign Language only involving the movements of fingers from a single hand as shown in Figure 11. Note that our system can be applied to other more complicated finger-level gestures on whichever wrist. We first conduct experiments in the static scenario (i.e., performing gestures without body movements), where each participant is asked to wear the prototype on the wrist of his/her dominant hand and perform the nine sign language gestures for 40 times respectively without any body movements. In addition, we conduct experiments in the body-motion scenario to study the impact of body movements on our system. To mimic the real-life scenarios where the participant may inevitably have motion when performing sign language gestures, we ask each participant to perform the same American sign language gesture (i.e., number 1) while performing body movements (i.e., swinging the forearm, rotating the forearm, swinging the body, and rotating the body). Each gesture is performed 10 times in one session. In particular, we have found that the sign gestures performed at different times have stable and similar waveform patterns, thus the thresholds we adopt are not affected over multiple sessions. In total, we collect over 7000 PPG segments and over 2500 motion sensor segments in the static and body-motion scenarios. Unless mentioned otherwise, our results are derived from 20 rounds Monte Carlo cross-validation using 50% of our data set for training the participant-dependent models and the rest for validation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2962760, IEEE Transactions on Mobile Computing

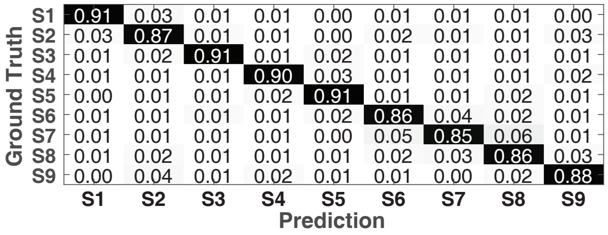
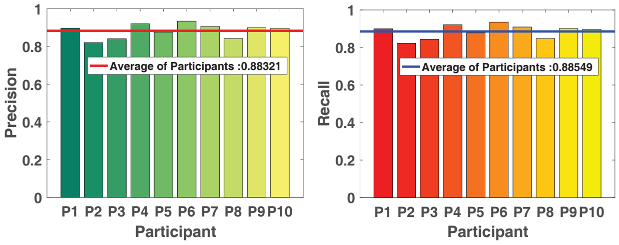


Fig. 12. Confusion matrix of recognizing nine sign language gestures among seven participants using PPG sensor.



(a) Average precision

(b) Average recall

Fig. 13. Impact factor study: comparison of gesture recognition performance among ten participants using PPG sensor.

8.1.3 Evaluation Metrics

Precision. Given N_g segments of a gesture type g, precision of recognizing the gesture type g is defined as $Precision_g = N_g^T/(N_g^T + M_g^F)$, where N_g^T is the number of gesture segments correctly recognized as the gesture g. M_g^F is the number of gesture segments corresponding to other gestures which are mistakenly recognized as the gesture type g.

Recall. Recall of the gesture type g is defined as the percentage of the segments that are correctly recognized as the gesture type g among all segments of the gesture type g, which is defined as $Recall_g = N_g^T/N_g$.

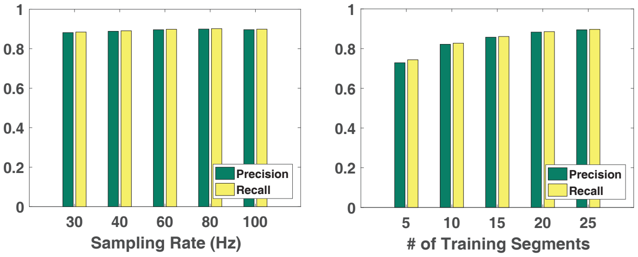
Confusion Matrix. In confusion matrix, each entry C_{ij} denotes the percentage of the number of gesture segments i was predicted as gesture type j in the total number of i. The diagonal entries show the average accuracy of recognizing each gesture, respectively.

8.2 Performance in Static Scenarios

8.2.1 GBT Performance Using PPG

First, we show the effectiveness of using the PPG sensor for sign language gesture recognition by evaluating the system performance when using the GBT classifier. It's important to note that we adopt the participant-dependent models, which means that our system trains the classifier for each user using his/her own data, respectively. Figure 12 depicts the confusion matrix for the recognition of the nine American Sign Language gestures with only using the PPG sensor. Specifically, the average accuracy is 88.32% with standard deviation 2.3% among all the 9 gestures. We find that the recognition results of the gesture $S2, S6, S7, S8$ are relatively low (i.e., around 86%). This is because those gestures have more subtle differences in the tendon/muscle dynamics than other gestures. Overall, the results confirm that it is promising to use commodity wrist-worn PPG sensors to perform sign language gesture recognition.

Impact of Different Users. Figure 13(a) and (b) present the average precision and recall of using PPG sensors to recognize each sign language gesture across different participants. We observe that all participants have high accuracy in recognizing these sign language gestures. Specifically, the average precision and recall of all the 10 participants are 88% and 89%, respectively, and the lowest average value of the precision and recall among all the participants is still above 80%. The results show the robustness and scalability of our proposed system across different users, and demonstrate the system is promising to act as an integrated function in commodity wearables once the interface of PPG raw signals to developers is open.



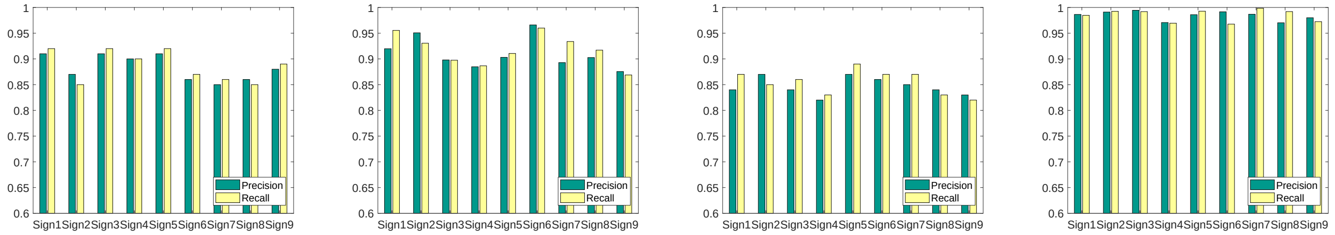
(a) Impact of sampling rate

(b) Impact of number of training segments

Fig. 14. Impact factor study: average precision and recall of recognizing nine gestures with different sampling rates and # of training segments using PPG sensor.

Impact of Sampling Rate. The sampling rate of sensing hardware is one of the critical impact factors on affecting the power consumption on wearables. We study the performance of the proposed system with different sampling rates on PPG sensors. Most of the commodity wearables have around $100Hz$ PPG sampling rate. For instance, Samsung Simband [43] configures its PPG sensor to $128Hz$ to perform time-centric tasks like Pulse Arrival Time calculations. In this study, we collect the PPG readings from the implemented wearable prototype with several sampling rates (i.e., $30Hz$ to $100Hz$ with step size 20Hz) to evaluate the system. Figure 14(a) shows the average precision and recall of the gesture recognition under different sampling rates. We find that the precision/recall increases with the increased sampling rate, however, the precision/recall still maintain as high as 87% under the lowest sampling rate (i.e., $30Hz$). As the results implied, our system is compatible with the off-the-shelf wearables with high recognition accuracy, and it can operate normally on the hardware with lower PPG sampling rate in terms of the power consumption.

Impact of Training Data Size. We change the amount of data used for training in the Monte Carlo cross-validation to study the performance of our system under different training data size as shown in Figure 14(b). In particular, we choose $5, 10, 15, 20$, and 25 PPG segments with respect to each gesture for training (i.e., $12.5\%, 25\%, 37.5\%, 50\%$, and 62.5% of our dataset) and use the rest of our data for validation, respectively. We observe that our system can achieve the average precision and recall of 89% and 90% respectively when 25 segments of training data for each gesture are collected in the training phase. As the size of the training data decreases, the system retains decent performance. Furthermore, the average precision and recall can achieve 75% and 77% respectively for recognizing nine sign langauage gestures using only 5 PPG segments of each gesture for training. The above results indicate our system can achieve good recognition performance with a limited size of training data (e.g., 5 sets per gesture), which ensures great convenience for practical usage on commodity wearables.



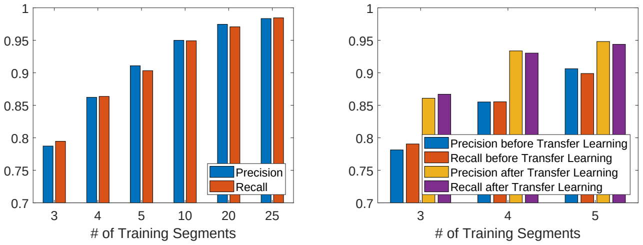
(a) PPG sensor only

(b) Gyroscope only

(c) Accelerometer only

(d) PPG + gyroscope + accelerometer

Fig. 15. Impact factor study: average precision and recall of recognizing nine gestures using GBT after integrating both PPG and motion sensor.



(a) Performance of ResNet with different training size

(b) Performance after applying transfer learning

Fig. 16. Performance of ResNet and transfer learning.

8.2.2 GBT Performance Using PPG and Motion Sensors

Next, we evaluate the performance of our system when motion sensors are available. Due to the human power limitation, we can't recruit the exact same 10 participants at the end of the project and only 7 out of 10 original participants are involved to conduct experiments with both PPG and motion sensors for a fair comparison. Figure 15 shows the comparison of the precision and recall of our system when 7 of 10 participants performing the nine sign language gestures wearing both PPG and motion sensors. In particular, we present the results in four cases that use different combinations of sensors, including 1) only PPG sensor; 2) only gyroscope; 3) only accelerometer; and 4) PPG, gyroscope and accelerometer, respectively. From Figure 15 (a), (b), and (c) we can observe that our system can achieve decent performance with the lowest precision and recall of over 83% and 84% respectively by using only one type of the sensors. In addition, we find that the performance of using the only accelerometer is not as good as that of using only the gyroscope. In particular, the precision of only using the accelerometer is 83%, which is lower than using the gyroscope sensor (i.e., 90%). The reason is that the sign language gestures do not involve a lot of displacements of the wrist and introduce more distinguishable rotations than accelerations in the wrist area. Moreover, as shown in Figure 15(d), when integrating PPG with both gyroscope and accelerometer, it is very encouraging to find that all those gestures can be recognized with the high average precision and recall of over 98%, which is over 10% more than the other three cases using only one type of the sensors. These results show that integrating the measurements from PPG and motion sensors can significantly improve the performance of our sign language gesture recognition.

8.2.3 Performance of DNN (ResNet) and Transfer Learning

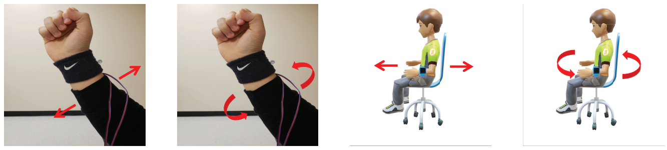
We also study the performance of our system with using the ResNet to build the underlying classifier. Since the training data size reflects the ease of use of the system in terms of the time for data collection, therefore we first study the performance of our system with different training data size as shown in Figure 16(a). In particular, we apply a shuffle split cross validation approach to randomly choose $3, 4, 5, 10, 20$, and 25 segments of each gesture for training respectively, and use the rest data for testing. We observe that our system achieves the similar performance as GBT with an average precision and recall of over 90% using 5 or more segments of each gesture in training. Moreover, our system retains decent performance with the decreasing training data size. The average precision and recall can still achieve to 78% when using only 3 segments for training each gesture.

In addition, since the specific architecture of the adopted ResNet can be taken advantage of by the transfer learning approach, we further explore the performance of our system after applying transfer learning. In particular, we first train a classifier for one person with using 25 segments of each gesture in order to generate the pretrained model that has excellent performance. Then we retrain the classifier with $3, 4$, and 5 segments of each gesture for the new people based on the pretrained model. As shown in Figure 16(b), we can see that after applying the transfer learning, the performance of our system is improved by 10% to 87% with only using 3 segments of each gesture for training. Moreover, with 4 or more segments of each gesture for training, the transfer learning can improve the precision and recall of our system to above 90%. Those results prove that the ResNet as the fundamental multivariate TSC is also suitable for our system, which not only provides the good performance without the procedure of extracting the features but also can take advantage of the transfer learning to significantly reduce the training effort to achieve decent performance.

8.3 Impact of Different Types of Body Movements

The wrist-worn PPG sensors basically monitor the blood flow in blood vessels on the wrist, which could possibly be impacted by the body movements that change the blood flow depending on how far the source of the body movements are away from the monitoring point. Therefore, we conduct the experiments with different types of body movements such as swing the forearm, rotate the forearm, swing the body, and rotate the body while performing the sign the language gesture as shown in Figure 17(a), (b), (c), (d) in order to explore their impacts respectively. In particular, we ask the participants to perform the gesture of American sign language number 1 (i.e., S1) while involving certain body movements. As shown in Figure 18 (a), the PPG gesture patterns of all the tested body movements have similar waveform shape as that of no body movement.

Additionally, we compare the DTW distances between the PPG segments collected during the time with and without body



(a) Swing forearm (b) Rotate forearm (c) Swing body (d) Rotate body

Fig. 17. Illustration of different types of body movements involved in the experiments.

movements to quantitively understand the similarity between the gesture-related patterns showed in Figure 18 (a). From Figure 18 (b) we can see that all the gesture-related patterns with body movements have small DTW distances of less than 900 to the gesture-related patterns without body movements, which indicates that the shapes of the PPG segments are well sustained as the case without any body movement. The small DTW distances suggest that our sign language gesture recognition system would work well by examining the PPG data that is insensitive to the body movements.

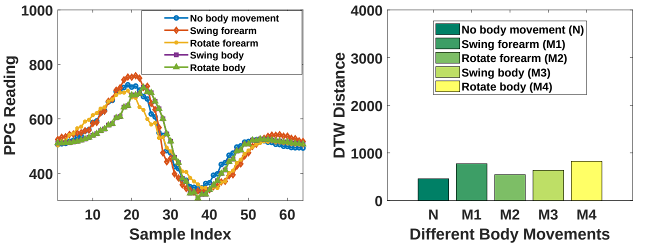
9 Discussion

Processing Delay. The processing delay of gesture recognition is critical to user experience in practical use. We develop an Android APP running on a Samsung Galaxy Note 5 (i.e., 1.5Ghz octa-core Exynos processor and 4GB of RAM) to track the elapsed time of major processing components (i.e., segmentation, feature extraction processes, and classification). We find that the processing delays are within a reasonable range which is about 0.651s and 0.601s for the GBT-based and ResNet-based approaches, respectively. The GBT-based approach has about 0.05s more in the processing delay than the ResNet-based approach because of the feature extraction process. We also notice that the processing delays of both approaches are dominated by the segmentation process (i.e., about 0.6s), especially the ending point detection in DTW-based approach, which takes about 0.41s. In addition, both GBT and ResNet have similar time for classification (i.e., 0.001s).

Energy Consumption. Our wearable prototype includes an Arduino board (i.e., about 50mA), two PPG sensors (i.e, 8mA) and one motion sensor (i.e., 4mA), respectively. In total, it is about 62mA current consumption of our prototype. Given the fact that the off-the-shelf smartwatches generally have a battery capacity of 380mAh, our system can run up to 6.1 hours on a smartwatch alone. If we offload the computation to a smartphone via the Bluetooth, the power consumption of the smartwatch [44] only involves the sensors and Bluetooth (i.e., 3.5mA), which is as low as 15.5mA. Given such low power consumption, our system can run over 24 hours on a smartwatch.

Skin Tone Impact. Humans have a diverse range of skin tones and different skin tones have different absorption of green light, impacting the gesture recognition accuracy. For example, darker skin absorbs more green light, limiting the capability to accurately measure heart rate. Using additional infrared LED PPG could mitigate the impact of skin tones.

Sensor Location Sensitivity. The location of the PPG sensor on the wrist is important, we carefully design it to be close to one of the main artery that can have more significant changes in blood flow when performing gestures. More sensors monitoring other arteries on the forearm could help increase the resolution of this sensing technology and facilitate more complicated finger-level gesture recognition, such as recognizing 26 letters in the American sign language.



(a) Raw PPG measurements (b) DTW distances

Fig. 18. Illustration of the impacts from different types of body movements on the PPG measurements when performing the same finger-level gesture (i.e., *S*1): (a) comparison of raw PPG measurements and (b) comparison of DTW distances that are collected when there are no body movements and different types of body movements.

Impact of Intense Body Movements. PPG sensor is sensitive to the extremely intense body movements. We find that some extremely intense body movements, such as intense wrist wringing and coughing, could induce large signal deformation than normal body movements (e.g., swing the forearm), and thereby affect the accuracy of wrist-worn PPG sensor readings. Therefore, it is recommended to not involve intense body movements when performing gesture recognition with our system.

Limitation and Future Work. Our work focuses on single-hand finger-level gesture recognition. We are aware that there are two-hand sign language gestures. In our future work, we will conduct new experiments with sign language involving two hands and explore the new features and methods (e.g., nature language processing techniques) for recognizing such two-hand sign language gestures. We are aware that using binary classifiers for each gesture is not the best solution. We find that transfer learning technology could use much less training effort to train a new model for a new user based on an old model of existing users. We will explore new approaches using the transfer learning technology to reduce the training effort for all the sign language gestures in our future work. And we will also explore the possibility of filtering the cardiac component from the motion artifacts (including both cardiac components and motion-related components) using the adaptive filter.

10 Conclusion

As an important means for human-computer interactions, gesture recognition has attracted significant research efforts in recent years. This paper serves as the first step towards a comprehensive understanding of the PPG-based gesture recognition with using motion sensors (i.e., accelerometer and gyroscope) as a complementary measure. We made a novel proposition to recognize the sign language gestures using low-cost PPG sensors and motion sensors in wearables. In particular, we develop a fine-grained data segmentation method that can successfully separate the unique gesture-related patterns from the PPG and motion sensor measurements. Additionally, we study the unique PPG and motion features resulted from finger-level gestures in different signal domains and devise a GBT-based system that can effectively recognize the sign language gestures by using PPG and motion sensor measurements. Moreover, we explore the deep neural network (ResNet) for classifying the multivariate time series signal (i.e., PPG and motion sensor measurements) and apply the transfer learning to significantly reduce the training effort. Our experiments with over 7000 PPG segments and 2500 motion sensor segments collected from 10 participants demonstrate that our system can differentiate nine elementary American Sign Language gestures with an average precision and recall over 89% with only using PPG sensor. We also reveal the limitation of using motion sensors alone and show that the sign language gesture recognition performance could be significantly improved by integrating the PPG and motion sensor data.

Acknowledgments

We sincerely thank Dr. Gokhan Ersan with the Department of Art in Binghamton University for creating the illustration figure. This work is supported in part by the National Science Foundation Grants CNS-1566455, CNS-1820624, and CNS-1514436.

References

[1] J. Klostermann Klostermann. (2015) Blinging up fashionable wearable technologies. [Online]. Available: https://cloudtweaks.com/2015/11/blinging-up-fashionable-wearable-technologies/

[2] A. Er-Rady, R. Faizi, R. O. H. Thami, and H. Housni, "Automatic sign language recognition: A survey," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2017, pp. 1–7.

[3] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE transactions on multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[4] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1565–1569.

[5] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 331–344.

[6] W. Mao, J. He, and L. Qiu, "Cat: high-precision acoustic motion tracking," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 69–81.

[7] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 1515–1525.

[8] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 77–89.

[9] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 27–38.

[10] P. Asadzadeh, L. Kulik, and E. Tanin, "Gesture recognition using rfid technology," *Personal and Ubiquitous Computing*, vol. 16, no. 3, pp. 225–234, 2012.

[11] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE transactions on human-machine systems*, vol. 44, no. 2, pp. 293–299, 2014.

[12] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 167–173.

[13] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[14] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM, 2015, pp. 9–14.

[15] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "Mole: Motion leaks through smartwatch sensors," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 155–166.

[16] M. Boukhechbaa, L. Caia, C. Wua, and L. E. Barnesa, "Actippg: Using deep neural networks for activity recognition from wrist-worn photoplethysmography (ppg) sensors."

[17] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Human activity recognition using accelerometer and photoplethysmographic signals," in *International Conference on Intelligent Decision Technologies*. Springer, 2017, pp. 53–62.

[18] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE transactions on multimedia*, vol. 17, no. 1, pp. 29–39, 2015.

[19] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14 991–15 015, 2016.

[20] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 1472–1480.

[21] S. Tan and J. Yang, "Wifinger: leveraging commodity wifi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2016, pp. 201–210.

[22] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 82–94.

[23] H. Wen, J. Ramos Rojas, and A. K. Dey, "Serendipity: Finger gesture recognition using an off-the-shelf smartwatch," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 3847–3851.

[24] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6425–6432, 2016.

[25] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiological measurement*, vol. 28, no. 3, p. R1, 2007.

[26] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, "Wearable photoplethysmographic sensors-past and present," *Electronics*, vol. 3, no. 2, pp. 282–302, 2014.

[27] INNERBODY.COM. (2017) Flexor digitorum superficialis muscle. [Online]. Available: http://www.innerbody.com/image_musc05/musc50.html

[28] T. Zhao, J. Liu, Y. Wang, H. Liu, and Y. Chen, "Ppg-based finger-level gesture recognition leveraging wearables," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1457–1465.

[29] ARDUINO. (2017) Arduino uno rev3. [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3

[30] T. Bombardini, V. Gemignani, E. Bianchini, L. Venneri, C. Petersen, E. Pasanisi, L. Pratali, D. Alonso-Rodriguez, M. Pianelli, F. Faita *et al.*, "Diastolic time–frequency relation in the stress echo lab: filling timing and flow at different heart rates," *Cardiovascular ultrasound*, vol. 6, no. 1, p. 15, 2008.

[31] E. Ebrahimzadeh and M. Pooyan, "Early detection of sudden cardiac death by using classical linear techniques and time-frequency methods on electrocardiogram signals," *Journal of Biomedical Science and Engineering*, vol. 4, no. 11, p. 699, 2011.

[32] B. Vuksanovic and M. Alhamdi, "Ar-based method for ecg classification and patient recognition," *International Journal of Biometrics and Bioinformatics (IJBB)*, vol. 7, no. 2, p. 74, 2013.

[33] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua science and technology*, vol. 19, no. 3, pp. 235–249, 2014.

[34] S. Noei, P. Ashtari, M. Jahed, and B. V. Vahdat, "Classification of eeg signals using the spatio-temporal feature selection via the elastic net," in *Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering (ICBME), 2016 23rd Iranian Conference on*. IEEE, 2016, pp. 232–236.

[35] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[36] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning new york," *NY: Springer*, pp. 115–163, 2001.

[37] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua, "Supervised feature learning for curvilinear structure segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2013, pp. 526–533.

[38] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[39] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[40] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *arXiv preprint arXiv:1809.04356*, 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] A. Ahmed. (1998) Frequency of a beating heart. [Online]. Available: https://hypertextbook.com/facts/1998/ArsheAhmed.shtml

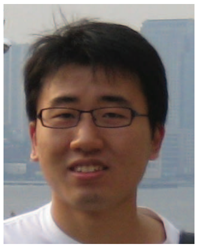
[43] Simband. (2017) Why is 128 hz used as a sampling frequency for the ppg signals? [Online]. Available: http://www.simband.io/documentation/faq.html

[44] X. Liu, T. Chen, F. Qian, Z. Guo, F. X. Lin, X. Wang, and K. Chen, "Characterizing smartwatch usage in the wild," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 385–398.

Tianming Zhao received his B.E. from the Department of Software Engineering, Chongqing University, China and his M.S. degrees from the Department of computer science, Binghamton University, USA. Currently, he is a Ph.D. student of the Computer & Information Sciences Department at Temple University. His research interests include mobile computing and sensing, cyber security and privacy, and smart health. He is currently working with Prof. Yan Wang.

Jian Liu is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville. He received his Ph.D. degree in Wireless Information Network Laboratory (WINLAB) at Rutgers University. His current research interests include mobile sensing and computing, cybersecurity and privacy, intelligent systems and machine learning. He is the recipient of the Best Paper Awards from IEEE SECON 2017 and IEEE CNS 2018. He also received Best-in-session Presentation Award from IEEE INFOCOM 2017, and two Best Poster Award Runner-up from ACM MobiCom 2016 and 2018.

Yan Wang is an Assitant Professor with Computer & Information Sciences Department at Temple University. Before that, he was with Department of Computer Science at SUNY, Binghamton. He received his Ph.D. degree in Electrical Engineering from Stevens Institute of Technology. His research interests include Cyber Security and Privacy, Internet of Things, Mobile and Pervasive Computing, and Smart Healthcare. His research is supported by the National Science Foundation (NSF). He is the recipient of the Best Paper Award from IEEE CNS 2018, IEEE SECON 2017 and ACM AsiaCCS 2016. He is the Winner of ACM MobiCom Student Research Competition, 2013. His research has been reported in numerous media outlets including MIT Technology Review, CNN, Fox News Channel, Wall Street Journal, National Public Radio, and IEEE Spectrum.



Hongbo Liu received the PhD degree in electrical engineering from the Stevens Institute of Technology. He has been at Indiana University Purdue University Indianapolis as an assistant professor in the Department of Computer Information and Graphics Technology since Aug. 2013. His research interests include mobile and pervasive computing, cyber security and privacy, and smart grid. He is the recipient of the Best Paper Award from ACM MobiCom 2011. Best Paper Runner-up Award from IEEE CNS 2013 and Best Paper Award from IEEE CNS 2018.

Yingying (Jennifer) Chen is a Professor of Electrical and Computer Engineering at Rutgers University and the Associate Director of Wireless Information Network Laboratory (WINLAB). She also leads the Data Analysis and Information Security (DAISY) Lab. Her research interests include smart healthcare, cyber security and privacy, Internet of Things, and mobile computing and sensing. She has co-authored three books, published over 150 journals and referred conference papers and obtained 8 patents. Her background is a combination of Computer Science, Computer Engineering and Physics. Prior to joining Rutgers, she was a tenured professor at Stevens Institute of Technology and had extensive industry experiences at Nokia (previously Alcatel-Lucent). She is the recipient of the NSF CAREER Award and Google Faculty Research Award. She also received NJ Inventors Hall of Fame Innovator Award. She is the recipient of five Best Paper Awards from IEEE CNS 2018, IEEE SECON 2017, ACM AsiaCCS 2016, IEEE CNS 2014 and ACM MobiCom 2011. She is the recipient of IEEE Region 1 Technological Innovation in Academic Award 2017; she also received the IEEE Outstanding Contribution Award from IEEE New Jersey Coast Section each year 2005 - 2009. Her research has been reported in numerous media outlets including MIT Technology Review, CNN, Fox News Channel, Wall Street Journal, National Public Radio and IEEE Spectrum. She has been serving/served on the editorial boards of IEEE Transactions on Mobile Computing (IEEE TMC), IEEE Transactions on Wireless Communications (IEEE TWireless), IEEE/ACM Transactions on Networking (IEEE/ACM ToN) and ACM Transactions on Privacy and Security.