Detecting Touch and Grasp Gestures Using a Wrist-Worn Optical and Inertial Sensing Network

Savannah Cofer, Tyler N. Chen, Jackie (Junrui) Yang, and Sean Follmer, *Member, IEEE*

***Abstract*—Freehand gesture-based interaction promises to enable rich interaction in applications such as augmented reality, virtual reality, human-robot interaction, and robotic prosthetic devices. However, current sensing approaches are limited to mid-air whole hand gestures and fail to identify small-scale tactile interactions with unsensorized environments. Detecting tactile interactions may unlock potential new applications in augmented reality and human-robot interaction in which unsensorized surfaces are used as touch input devices. This work presents a novel wrist-worn sensing device that combines near-infrared and inertial measurement unit sensing to enable high-accuracy detection of surface touch and grasp interactions. Two convolutional neural networks were used to map device inputs to detect touch events, and classify them by gesture type or direction. We evaluated the accuracy and temporal precision of our system for event detection and classification. Results from an in-lab user study of 12 participants showed an average of 97% touch detection accuracy and 98% grasp detection accuracy. In our study, we found that near-infrared and inertial sensing are complementary and can be used in tandem to effectively address both touch event detection and directionality classification.**

***Index Terms*—Virtual reality and interfaces, human detection and tracking, touch in HRI.**

I. Introduction

SINCE the invention of the capacitive touchscreen, smartphones and tablets have provided a popular means of human-computer interaction (HCI), enabled by a consistent language of high-precision surface touch gestures. Augmented and virtual reality (AR/VR) interfaces promise to unlock new HCI and human-robot interaction (HRI) paradigms beyond the limitations imposed by 2D screens [1]–[4]. However, current AR/VR interactions are mostly limited to low-precision mid-air gestures, and input devices lack the ability to detect touch microgestures on unsensorized objects and surfaces [5], [6]. From a usability standpoint, it is known that digital interaction is difficult without tactile feedback from a physical haptic surface [6], [7]. This motivates the development of methods to reliably detect physical surface touch and object grasp in AR and VR [8], [9]. While some commercial AR/VR devices currently use camera-based hand tracking, this method has difficulty detecting surface contact interactions due to occlusion, depth-estimation error, and high computational requirements [10]–[12]. Other devices rely on hand-held controllers, which prevent tactile interaction with real objects altogether. While a variety of wearable approaches have demonstrated detection of mid-air gestures made by the whole hand, few have been extended to the difficult problem of detecting small-scale tactile interactions such as those used with touchscreen devices (Table I) [13]–[15]. Thus, there is an unmet need within AR systems for reliable hands-free sensing of interactions with tangible environments, which may enable new interfaces controlled by smartphone-like surface touch gestures on unsensorized surfaces [8], [9], [11].

Prior work on surface touch and grasp detection (Table I) primarily used peripheral sensing devices mounted to a user's wrist or fingers to detect interactions between the hand and a surface. Many of these freehand gesture detection systems utilized surface electromyography (sEMG) and force myography (FMG) devices to measure muscle tension in a user's forearm. These methods exhibit high detection accuracy, but the results are highly sensitive to sensor-skin coupling and require extensive calibration [16]–[21]. Other systems utilized inertial measurement unit (IMU) sensing at the fingertips for touch gesture classification, but a fingertip-mounted sensor may limit potential applications that require unimpeded hand motion [22], [23].

Beyond the sensing methods described above, the use of near-infrared (NIR) sensing to measure tissue displacement in the wrist offers a promising alternative for gesture control. Previous work indicated that NIR sensing offers a high degree of robustness to changes in sensor-skin coupling, and can be calibrated to account for a device's position on the user's wrist [13], [24], [25]. While NIR sensing has been used to robustly identify mid-air whole hand gestures, the potential of wearable NIR sensing for detection of fine-grained surface contact interactions remains unexplored [13], [26], [27]. To our knowledge, detection of smartphone-like interactions on a passive tabletop surface has not been demonstrated with a wrist worn NIR device that senses motion within the user's body (as opposed to the user's surroundings). In this work, we sought to explore whether a

Manuscript received 24 February 2022; accepted 18 June 2022. Date of publication 15 July 2022; date of current version 23 August 2022. This letter was recommended for publication by Associate Editor S. F. Atashzar and Editor J.-H. Ryu upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation Graduate Research Fellowship (S.C. and T.N.C.), in part by the Knight-Hennessy Fellowship (S.C. and T.N.C.), and in part by the Alfred P. Sloan Research Fellowship under Grant FG-2021-15851. *(Savannah Cofer and Tyler N. Chen contributed equally to this work.) (Corresponding author: Sean Follmer.)*

Savannah Cofer and Sean Follmer are with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: cofer@stanford.edu; sfollmer@stanford.edu).

Tyler N. Chen is with the Department of Bioengineering, Stanford University, Stanford, CA 94305 USA (e-mail: tnchen@stanford.edu).

Jackie (Junrui) Yang is with the Department of Computer Science, Stanford University, Stanford, CA 94305 USA (e-mail: jackiey@stanford.edu).

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of Stanford University under Application No. 34826.

Digital Object Identifier 10.1109/LRA.2022.3191173

TABLE I
SUMMARY OF RELATED WORK FOR GRASP AND TOUCH DETECTION

Name	Sensing Input Method	Algorithm	Reported Performance Metric (n participants)
Xiao (2017) [28]	2 FMG straps on wrist and forearm, 8 sensors	Linear Discriminant Analysis (LDA)	95% grasp detection accuracy from wrist, 91% from forearm (n = 10)
Jiang (2018) [20]	FSR wristband with 16 sensors, FSR on palm of hand	Linear Discriminant Analysis (LDA)	82% average classification accuracy for 16 grasp events (n = 9)
Zhang (2019) [29]	On-skin touch segmentation using computer vision and RF sensing	Fourier transform with threshold	93.8% touch segmentation accuracy (n = 10)
Becker (2018) [15]	EMG using Myo armband	LSTM Recurrent Neural Network	97% touch detection accuracy from forearm (n = 18)
Shi (2020) [22]	Single IMU mounted at fingertip	2D Convolutional Neural Network	95% touch detection acc. with fingertip IMU, 82.6% with IMU on proximal phalange (n = 12).
Xiao (2018) [11]	Microsoft HoloLens camera	Random sample consensus (RANSAC) computer vision	96.5% touch detection accuracy - 3.5% missed touches and 19% spurious extra touches (n = 17)
Cofer and Chen (2022)	**NIR + IMU Sensing Network on wristband**	**Moving Window 1D CNN in Two-Level Architecture**	**97% touch detection accuracy, 98% grasp detection accuracy (n = 12)**

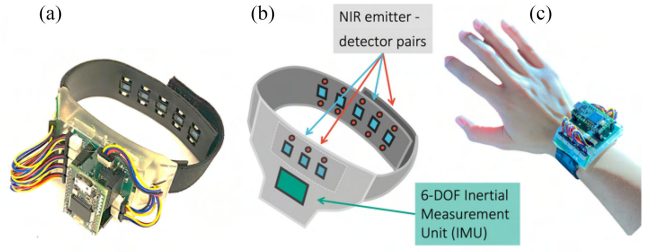

Fig. 1. (a) Gesture detection device shown with around-the-wrist NIR sensing and IMU. (b) NIR+IMU schematic. (c) Device worn on user.

multimodal device combining NIR and IMU sensing could enable high-accuracy detection of object grasp and smartphone-like surface touch while maintaining a wristwatch form factor. Motivated by the goal of enabling tactile interaction with passive surfaces, we assess our wrist-worn device's performance for the following objectives:

Objective 1: Detect whether a user is grasping an object, including the duration of the grasp.

Objective 2: Detect whether a user's finger is in contact with a surface.

Objective 3: Classify touch interaction gesture types, including Tap, Hold, Swipe, Scroll, and Zoom gestures.

In this work, we first designed and fabricated the data collection device in Fig. 1 and recorded synchronized sensor data and ground truth labels for contact. Second, we conducted a user study with 12 participants who completed a set of object grasping and surface interaction tasks while wearing the device. Third, we developed a data processing and Machine Learning (ML) pipeline to identify and classify grasp and touch events, including evaluation of Logistic Regression (LR), Naive Bayes classifiers (NB), Multi-Layer Perceptron (MLP), One Dimensional Convolutional Neural Network (1D CNN), General Regression Neural Network (GRNN) and Linear Discriminant Analysis (LDA) algorithms. Our work demonstrates the potential of a wrist-worn NIR + IMU input device for grasp and touch detection, which may enable future applications in AR/VR systems and human-robot interaction.

II. GESTURE DETECTION DEVICE

The gesture detection device was designed to be worn like a watch, and continuously collects optical and inertial sensor data during usage. The device uses single-wavelength (940 nm) infrared emitters and NIR-sensitive photodiodes to record the optical intensity of scattered light from the top few millimeters of the user's skin [30]. There are three emitter-receiver pairs located on the dorsal side of the wrist and five on the ventral side, based on previous work which found high accuracy levels of mid-air gesture recognition using an around-the-wrist NIR configuration [13]. The device pulses each NIR emitter in turn while receiving data from its nearest-neighbor photodiodes, resulting in a total of 20 incoming data channels, with 7 channels on the dorsal side of the wrist and 13 on the ventral side. An analog front end controller is used to drive and sample each of the 20 active LED-photodiode pairs every 7.5 ms for a refresh rate of 133 Hz. This rate was selected to strike a balance between temporal sensitivity to optical changes at the wrist and an ability to reject noise with a longer integration window. The device also contains an ICM20948 inertial measurement unit (IMU) mounted to the top side of the wrist, which samples accelerometer and gyroscope values at 225 Hz. A Teensy3.2 microcontroller (PJRC) is used for power supply and tethered serial output. LED-photodiode pairs on the ventral side of the wrist are mounted on a flexible printed circuit board (PCB) in rubber casing with adjustable hook-and -loop fasteners to conform to the shape of the user's wrist and accommodate a diverse set of participants.

III. DATA COLLECTION

A. User Study Task Design

We conducted an in-lab user study with twelve participants approved by the Institutional Review Board of Stanford University (IRB #34826) to investigate the performance of NIR and IMU sensing from a wrist-worn device in conjunction with machine learning algorithms to detect physical grasp and surface contact events. Trials consisted of two parts – grasp and touch tasks. Conductive contact between a finger-worn copper mesh and copper plate provided ground truth contact data which was synchronized with sensor data prior to transmission via the serial output channel on the wrist-worn device.

For the grasp section of the study, participants were asked to grasp a static object without lifting, shown in Fig. 2, and pick up and move an object to a new target, shown in Fig. 3. The target object for grasp and pickup trials was a polylactic acid (PLA)

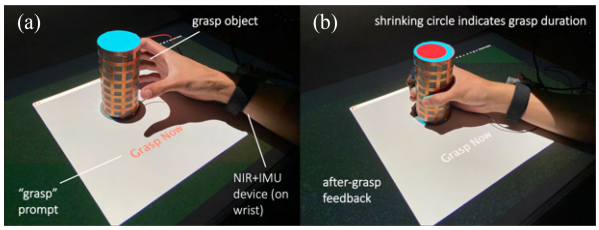

Fig. 2. (a) Static grasp indicated by "Grasp Now" display. (b) The duration of static grasp indicated by shrinking red target circle.

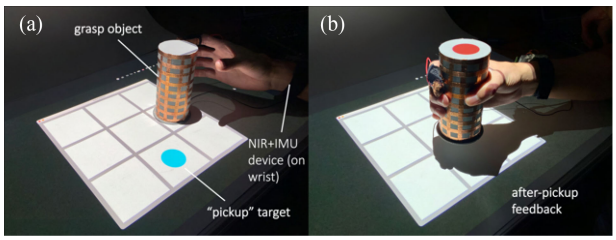

Fig. 3. (a) Grasp and pickup location indicated by blue target circle. (b) Target circle turns red once object is grasped.

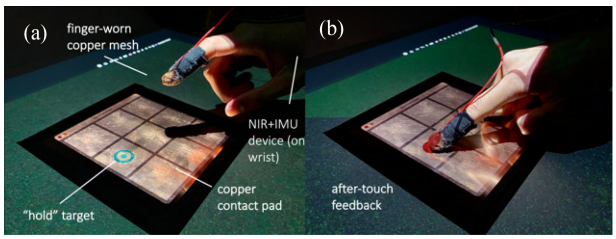

Fig. 4. (a) Index finger hovering in air above "hold" target with fixed hand posture. (b) Index finger retaining the same hand posture and contacting target.

cylinder (h = 12 cm, d = 6 cm) as used by Xiao et al. [28], which is the average object diameter for the Large Diameter Thumb Abducted Grasp from the GRASP Taxonomy of Feix et al. [31]. Grasp event duration was randomized from 1 to 3 seconds in the static grasp trial, shown in Fig. 2. The total target space for the grasp and pickup trials was 60 x 60 cm, with 20 cm between targets as shown in Fig. 3.

For the touch interaction tasks, participants were asked to complete basic touch interaction gestures based on those supported by the Universal Windows Platform (UWP) [32] and Apple iOS [33], as shown in Figs. 4 and 5. Tasks included Tap, Hold, Swipe, Scroll, and Zoom gestures, as well as a Hybrid task that included all five. The total target space for these touch tasks was 30 x 30 cm, with 10 cm between targets. We used the standard web content accessibility guidelines (WCAG 2.2) for pointer target spacing to ensure that the target size and spacing was sufficient [34]. The target location was ordered within a 3x3 grid using block randomization, such that each location was touched an equal number of times but in randomized order. For the Swipe, Scroll, and Zoom gestures, the gesture directionality was also ordered using block randomization. The onset of press

GESTURE	DESCRIPTION
	TAP Touch finger lightly to target. Each user performs 36 taps within a 3x3 grid (8 taps per square, randomized order).
	HOLD Touch and hold for a two second duration as target shrinks. Each user performs 36 taps within a 3x3 grid (8 taps per square, randomized order).
	SWIPE Move finger quickly in arrow direction. Each user performs 72 swipes within 3x3 grid (18 swipes in each up, down, left, and right directions).
	SCROLL Move finger from one target to another over a two second duration. Each user performs 72 swipes (18 swipes in each up, down, left, and right directions).
	ZOOM Move two fingers closer together for pinch, farther apart for expand. Each user performs 36 zoom gestures (18 pinch or expand per direction).

Fig. 5. Set of touch based interaction gestures based on those supported by the Universal Windows Platform (UWP) and Apple iOS [32][33].

was uniformly distributed within a 2 s window to avoid effects of periodicity.

B. Participants

In order to ensure data collection from multiple demographics, 12 participants self-reported gender, race and ethnicity, age, and wrist diameter. Participants self-reported as 6 male (50%) and 6 female (50%), including 3 East or Southeast Asian (25%), 2 Black or African American (16.7%), 3 South Asian or Indian Subcontinent (25%), and 4 White (33.3%). Participants ranged in age from 23 to 53 years, with a mean of 27.17 years and median of 24.5 years (standard deviation = 8.58 years). Measured wrist diameter ranged from 14 to 19.5 cm, with a mean of 16.35 cm and median of 15.88 cm (standard deviation = 1.71 cm). All participants were right-handed.

C. Procedure

Participants sat at a horizontal table with their elbow suspended midair and were asked to extend their arm to contact the farthest target, displayed by an overhead projector. For each task, a demonstration video was displayed on screen and three training trials were conducted before data collection. For the grasp trials, users completed 36 static grasps of varying duration (1-3 seconds) followed by 72 object pickups. For the touch interaction gesture set, each user first completed a set of 36 Tap gestures, followed by 36 Hold, 72 Swipe (18 per direction), 72 Scroll (18 per direction), and 36 Zoom gestures (18 per direction). The gesture directionality and timing of the target prompts were randomized within each gesture set. Next, users completed 90 touch gestures in randomized order, consisting of

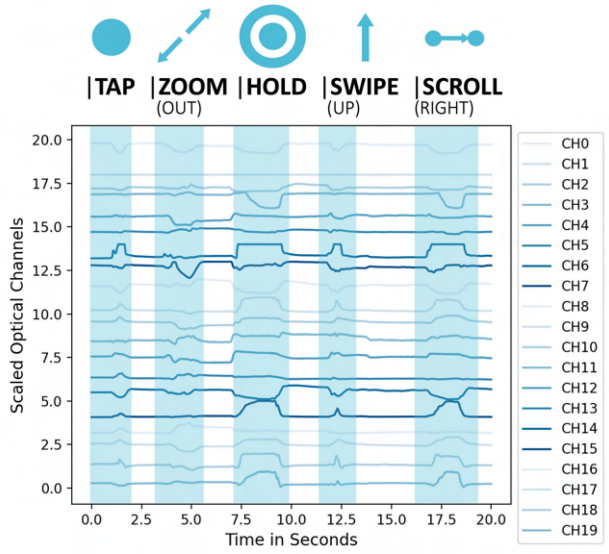

Fig. 6. Representative sample of 20 NIR data channels for touch interaction gestures from a single user.

18 repetitions of each of the 5 gesture types. Total active testing time was 60 minutes per user, and users were provided with break time between each set.

IV. Methods

The primary goal of our data processing pipeline is to detect and classify surface touch and object grasp events using wrist-worn sensing by training on ground truth conductive contact. To do this, we took a supervised learning approach, where the input training set was composed of wrist-worn NIR and IMU data labeled with ground truth conductive contact data, and the output testing set was composed of wrist-worn NIR and IMU data only. A representative sample of the NIR data from all 20 channels is shown in Fig. 6. To preprocess the NIR data, we applied a 1D Savitzky-Golay filter across time reduce the effects of noise, scaled input data to zero mean and unit variance, and randomly scrambled the input dataset with a 500 datapoint (4 s) window to reduce the effects of periodicity. For the IMU data, we used the derivative of accelerometer data to eliminate the offset caused by gravity.

A. Selection of Classification Algorithm

Based on prior work for touch and gesture detection [11], [18]–[20], [22], [28], [29], [35], we initially evaluated the performance of multiple ML algorithms for binary event detection from our sensor data, as summarized in Table II. To implement MLP, LDA, LR, and NB, we used the scikit-learn package [36]. To implement GRNN and CNN, we used the NeuPy and TensorFlow packages, respectively [37], [38].

Based on this initial comparison, we proceeded to use the Timeseries One-Dimensional Convolutional Neural Network (1D CNN) for binary and multi-output classification for the results presented in Section V: A-C. 1D CNNs with timeseries windowing have shown prior success for sequential data [38].

TABLE II
Comparison of ML Algorithms for Touch Event Detection

Algorithm	Average Accuracy (%)	SD Across Users
Multilayer Perceptron (MLP)	95.37%	0.1238
Gen. Regression Neural Net. (GRNN)	74.58%	0.3010
Lin. Discriminant Analysis (LDA)	97.60%	0.0407
Logistic Regression (LR)	96.80%	0.0788
Gaussian Naive Bayes (GNB)	93.79%	0.1418
Bernoulli Naive Bayes (BNB)	93.70%	0.1584
Timeseries 1D CNN (Window = 11)	**98.48%**	**0.0503**

TABLE III
NIR Vs. IMU for Contact Detection

Task	NIR Only	IMU Only	NIR + IMU
Tap	97.14%	86.00%	94.32%
Hold	97.20%	91.46%	97.69%
Swipe	99.65%	94.99%	97.61%
Scroll	97.56%	82.40%	99.17%
Zoom	95.64%	90.74%	97.11%
Average	**97.44%**	**89.12%**	**97.18%**

TABLE IV
NIR Vs. IMU for Directionality Classification

Task	NIR Only	IMU Only	NIR + IMU
Swipe Direction	41.90%	97.48%	47.63%
Scroll Direction	30.88%	96.18%	36.60%
Zoom (Expand/Pinch)	88.05%	97.60%	92.05%
Average	**53.61%**	**97.09%**	**58.76%**

The selected model was a variant of the LeNet framework [39]. Based on a hyperparameter sweep over layer size, kernel size, windowing size, regularization rate, pooling layer size, model depth, and number of epochs, our touch event detection architecture uses the following structure:

- Input: 20 channels of timeseries data (window size = 11)
- 1D CNN layer with 24 filters, kernel size = 5, Rectified Linear Unit (ReLU) activation, L2 Regularization
- Max pooling layer (pool size = 2), Batch Normalization
- 1D CNN layer with 16 filters, kernel size = 5, ReLU activation, L2 Regularization, Flattening layer
- Dense layer of size 30 with ReLU activation function
- Dense layer of size 2, SoftMax activation and conversion to binary output with threshold 0.2
- Compilation with sparse categorical cross entropy loss and Adam optimization.

B. Detection and Classification Pipeline

We separated the tasks of touch detection, gesture type classification, and direction classification into separate neural networks. Based on the results summarized in Tables III and IV in *Section V. Experimental Results*, we propose an architecture with two sequential 1D CNNs as shown in Fig. 7 to detect and classify touch events. The first is a user- and trial-specific NIR model to detect and segment touch events, which is followed by a cross-user model utilizing IMU data for gesture classification. In this architecture, raw NIR data is preprocessed, split by a moving timeseries window, and used as an input to a 1D CNN which

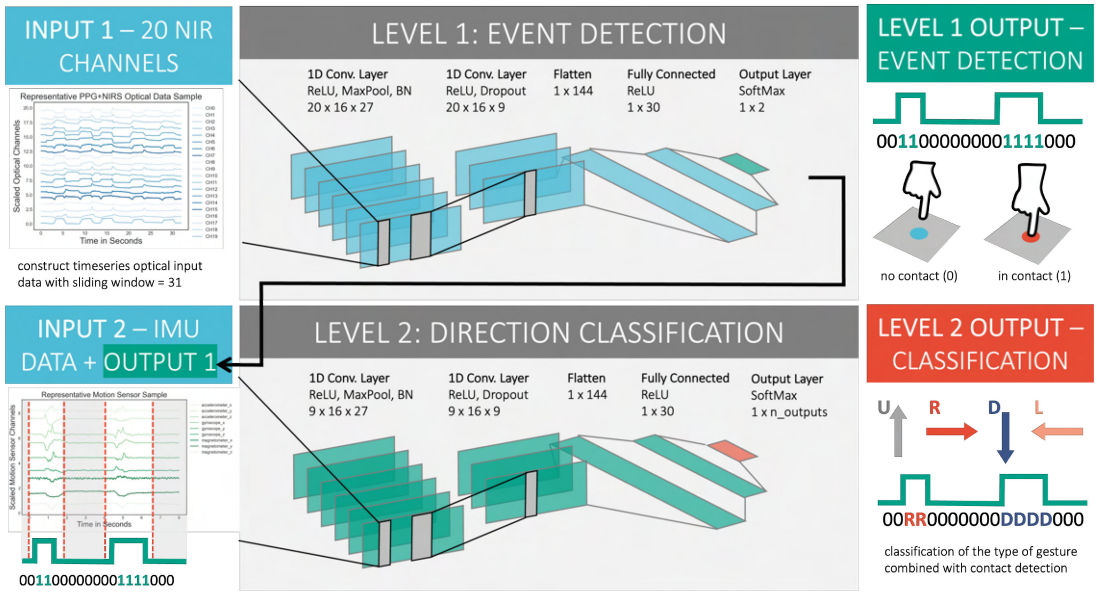

Fig. 7. Two-level architecture for touch detection and classification. 20 NIR channels are used to detect surface touch on the first level. The resulting binary event sequence is then used to construct a timeseries IMU dataset for classification of event directionality on the second level, given the type of gesture (Swipe, Scroll, or Zoom). All raw input data is preprocessed and fed through a timeseries moving window prior to training through the 1D CNN.

identifies whether a user's hand is in contact with the surface over time. This Level 1 output, generated using NIR data, is used to construct the Level 2 IMU dataset (gyroscope + accelerometer), which is fed into the second 1D CNN to accurately classify the type of gesture performed. For the purposes of this study, the Level 1 and Level 2 models were trained separately on ground truth labels rather than being trained end-to-end in order to evaluate each task individually. IMU data across all users is selected as training data for the Level 2 model due to its high generalizability across users, in contrast to the NIR data, which in this study was used in a user- and trial-specific Level 1 model. Level 2 of the architecture completes the time-dependent gesture identification by classifying each individual gesture before integrating with the results from Level 1.

C. Event Recognition Metrics

In this work, we report the percent of correct predictions, missed events, and spurious predictions as shown in Fig. 8. When a change from low to high occurs in either the predictions or ground truth time series, we check whether the other time series encounters a corresponding increase from low to high before the initial time series returns to the low state. If such an increase is detected, the event is correctly detected. If not, the event is considered a False Negative if it is detected initially in the ground truth time series, or a False Positive if detected initially in the prediction time series.

After producing a binary array of predictions p, where $p(t) = 1$ if an event is detected and $p(t) = 0$ if no event is detected, we obtain a predicted output that occasionally introduces spurious oscillations between the 0 and 1 state. We use our assumption of a minimal press length $L = 0.16\,s$ where each pulse of high and low output is at least length L. This means that if a signal oscillates too quickly, it is dropped as an instability.

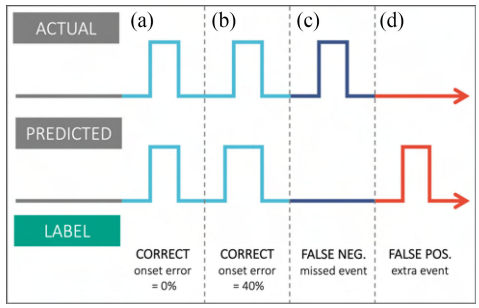

Fig. 8. Demonstration of binary event detection metrics between actual and predicted events, including. (a) correct detection with 0% onset error. (b) correct detection with 40% onset error. (c) false negative (missed event). (d) false positive (spurious touch).

V. Experimental Results

A. Objective 1: Grasp and Release Detection

We first processed the data from the grasp and pickup trials to see whether a model trained on NIR sensing data could accurately detect whole-hand grasp and pickup events. Based on our comparison of the classification algorithms in Table II, we trained a timeseries 1D CNN on each of the grasp and pickup trails, resulting in a mean static grasp accuracy of 97.62% (SD = 5.3%) across 12 users. Pickup events were detected with a mean accuracy of 98.95% (SD = 4.9%). A moving window size of 301 timesteps (approximately 2.4 s) was selected to optimize for the duration of grasp events. For grasp event prediction, the model was retrained separately for each user.

The length of each grasp event varied uniformly between 1 to 3 seconds. Fig. 9 shows the error in the prediction of the onset (grasp) and offset (release) actions for the static grasp trial, where the total length of each grasp is normalized to 1.

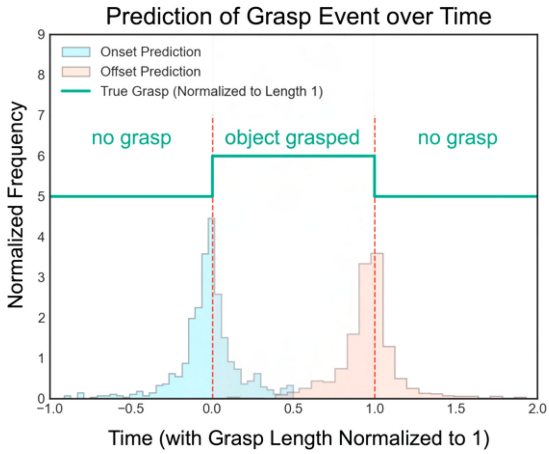

Fig. 9. Prediction of grasp event over time, with the length of each grasp event normalized to 1 and n = 672 predicted grasp events across 12 users with 5-fold cross validation.

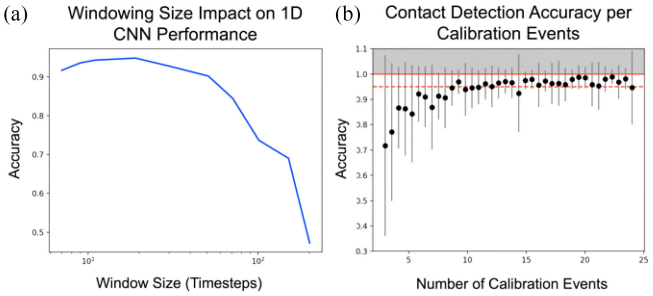

Fig. 10. (a) 1D CNN performance for Tap detection is maximized at window = 11 timesteps, or approximately 83 milliseconds. (b) Effect of number of calibration (training) events on testing accuracy for contact detection while using timeseries 1D CNN across 12 users.

Root mean square (RMS) onset error was 228 ms and offset error was 287 ms on a per-event basis.

B. Objective 2: Touch Event Detection

Based on results from *Objective 1: Grasp and Release Detection*, we retrained the 1D CNN model and evaluated its performance for detection of smartphone-like finger touch events (Tap, Hold, Scroll, Swipe, and Zoom) on a passive tabletop surface. We conducted a hyperparameter sweep to optimize layer size, kernel size, windowing size, regularization rate, pooling layer size, model depth, and number of epochs for each gesture trial. For example, Tap detection accuracy was maximized for a windowing size of approximately 0.1 seconds as shown in Fig. 10 a. These results form Level 1 of the 1D CNN architecture illustrated in Fig. 7. Fig. 10 b shows the effect of increasing the number of calibration events on total event detection accuracy using this timeseries 1D CNN, where >95% event detection accuracy is reached at >10 calibration events. Using the per-user model shown in Level 1 of Fig. 7 while training on 10 calibration events per gesture on each user, we were able to achieve >95% event detection accuracy for each gesture type averaged across 12 users. Fig. 11 illustrates the performance of the 1D CNN

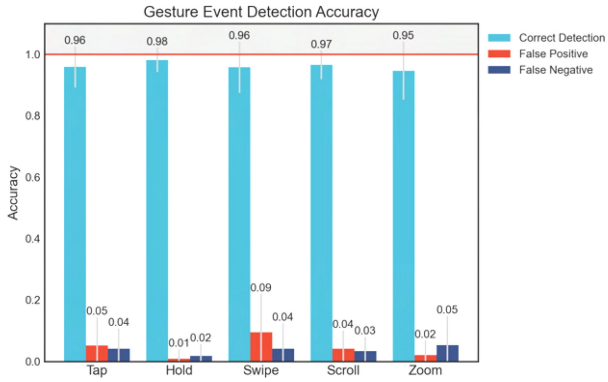

Fig. 11. >95% touch interaction gesture event detection accuracy (NIR only) for all gestures averaged across 12 users.

across different types of touch detection tasks as described in *Section II. Data Collection* and summarizes both the overall event detection accuracy and prevalence of missed events (false negative) and spurious touches (false positive) as defined in Fig. 8. The simple non-directional Hold (single finger, 2 s press) gesture demonstrated the highest accuracy at 98% across all users and <1% spurious touches, while dynamic touch gestures with multiple subclasses experienced higher error rates.

C. Objective 3: Touch Gesture Classification

We sought to evaluate whether the approach reported in *Objective 2: Touch Event Detection* might be extended to the task of direction classification by training on detected touch events. To do this, we first compared the effects of training on NIR data only, IMU data only, and combined NIR and IMU data for contact detection and directionality classification in Tables III and IV respectively. We found that contact detection (Level 1) was best achieved using NIR data with an average of 97.44% accuracy, while IMU alone only reached 89.12% accuracy. For subsequent directionality classification (Level 2), however, we found that the opposite was true - directionality of gestures was best classified using IMU sensing with 97.09% accuracy, whereas NIR models failed to work robustly. This may be due to the cross-user generalizability of the IMU data, which allowed us to train the model on data from all users rather than retraining for each user. This multi-user 1D CNN trained on IMU data forms Level 2 of our data processing architecture.

To generate this result, we segment our data by detected touch events and type, and classify the directionality of Swipe, Scroll, and Zoom events by training and testing on IMU data across all users. Swipe directionality classification (shown in Fig. 12) was achieved with an accuracy of 96.6% Up, 95.3% Down, 98.9% Right, 99.1% Left. Scroll directionality classification performed similarly well, with an accuracy of 97.4% Up, 96.8% Down, 95.1% Right, 95.4% Left. Zoom directionality classification was achieved with an accuracy of 99.3% Expand, 95.9% Pinch. We also used Level 2 of the data processing architecture from Fig. 7 to classify the type of gesture detected. To improve robustness, we first trained on IMU data collected from all users' single-interaction trials, then refined the model on the data from

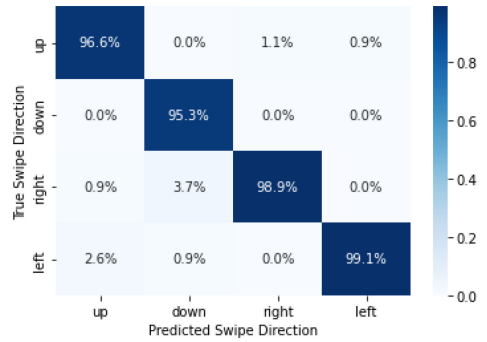

Fig. 12. Swipe event directionality classification confusion matrix. Scroll and Zoom results reported in text.

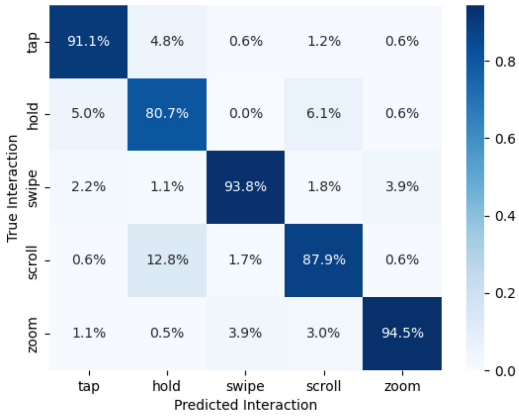

Fig. 13. Interaction gesture type classification confusion matrix between Tap, Hold, Swipe, Scroll, and Zoom events.

a particular user's hybrid trial where they completed 18 of each of the 5 touch interaction gestures for a total of 90 gestures per user. Gesture type classification was achieved with 91.1% Tap, 80.7% Hold, 93.8% Swipe, 87.9% Scroll, and 94.5% Zoom.

VI. Discussion

In this study, we sought to evaluate whether a wrist-worn device utilizing a combination of NIR and IMU sensors could achieve high-accuracy detection of tactile interactions with passive surfaces and objects. For grasp detection, we found that a neural network trained on NIR data detected 1 to 3 s grasp events with < 300 ms RMS error for grasp onset and release, suggesting the potential for real-world object manipulation in augmented reality and human-robot collaborative environments. When assessing performance across our touch detection objectives, we find that NIR and IMU data are highly complementary. Whereas 20-channel NIR sensing served as an accurate event detector for each individual user (Table III), the IMU provided generalizable data for cross-user classification of gesture subtypes (Fig. 13) and direction (Fig. 12 and Table IV) after a touch was detected. We found that wrist-worn NIR sensing was highly effective and outperformed IMU-only models for hands-free detection of tactile interactions with nonsensorized surfaces and objects, with 97% mean grasp and surface touch detection accuracy on a diverse cohort of users and multiple gesture types (see Table III). While an IMU-only model underperformed for contact detection, it was highly effective for classification of gesture subtypes (Fig. 13) and direction (Fig. 12) across users. By conditionally training the IMU classification model on distinct touch events identified by the NIR model, the combined models are able to distinguish between UWP + iOS touch gestures on nonsensorized surfaces. However, NIR sensing alone is not without its limitations. Due to differing wrist geometries and sensor placements between users, a new CNN model was trained for each contact detection trial, indicating the need for retraining or calibration of NIR-only models prior to use in HCI and HRI applications. We found that this initial calibration phase only required 10 touch interactions (Fig. 9 b) to achieve comparable performance to the fully-trained model for a new user and a new touch gesture. The NIR + IMU model evaluated in Table IV had variable classification performance across touch gestures, indicating the need for improved model tuning for mixed-modality inputs.

VII. Conclusion and Future Work

In this study, we present a novel NIR + IMU sensing device that achieved 97% touch detection accuracy and 98% grasp detection accuracy, with 96% classification accuracy for gesture directionality and 90% classification accuracy between five touch gesture types using convolutional neural networks. These results are comparable to or exceed those of prior literature (Table I) and leverage two complementary sensing modalities to accurately classify industry-standard touch gestures on nonsensorized surfaces. In future work, we plan to train our model on a larger dataset of input gestures such that touch event identification, gesture classification, and directionality classification can be performed sequentially with minimal latency. We further plan to develop multi-user ML models which include calibration to account for physical variations in device position. To reduce the need for upfront calibration, we expect that future applications based on this modality can utilize high-confidence labels from the IMU to calibrate and train the NIR model in real time for event detection and to correct for signal drift due to device motion across multiple sessions. Another ongoing area of development is evaluating the performance trade-offs between model accuracy, computation time, and prediction latency, by considering more lightweight ML algorithms such as LDA and LR that performed nearly as well as the 1D CNN (Table II). Ultimately, we plan to integrate these improvements and evaluate the performance of our device in targeted application spaces, specifically enabling novel interactions in augmented reality and in upper-limb robotic prosthetic devices.

References

[1] I. Radu, "Augmented reality in education: A meta-review and cross-media analysis," *Pers. Ubiquitous Comput.*, vol. 18, pp. 1533–1543, 2014.
[2] D. R. Berryman, "Augmented reality: A review," *Med. Reference Serv. Quart.*, vol. 31, no. 2, pp. 212–218, 2012.
[3] M. Desselle et al., "Augmented and virtual reality in surgery," *Comput. Sci. Eng.*, vol. 22, no. 3, pp. 18–26, 2020.

[4] E. Bottani and G. Vignali, "Augmented reality technology in the manufacturing industry: A review of the last decade," *IISE Trans.*, vol. 51, no. 3, pp. 284–310, 2019.
[5] K. Wolf et al., "A taxonomy of microinteractions: Defining microgestures based on ergonomic and scenario-dependent requirements," in *Proc. IFIP Conf. Hum.-Comput. Interact*. 2011, pp. 559–575.
[6] B. Ens et al., "Exploring mixed-scale gesture interaction," in *Proc. SIGGRAPH Asia Posters*, 2017, pp. 1–2.
[7] M. R. Jakobsen et al., "Should I stay or should i go? Selecting between touch and mid-air gestures for large-display interaction," in *Proc. IFIP Conf. Hum.-Comput. Interact.* 2015, pp. 455–473.
[8] E. Goh et al., "3D object manipulation techniques in handheld mobile augmented reality interface: A review," *IEEE Access*, vol. 7, pp. 40581–40 601, 2019.
[9] S. M. Nizam et al., "A review of multimodal interaction technique in augmented reality environment," *Int. J. Adv. Sci., Eng. Inf. Tech.*, vol. 8, pp. 1460–1469, Sep. 2018.
[10] L. Kugler, "The state of virtual reality hardware," *Commun. ACM*, vol. 64, no. 2, pp. 15–16, 2021.
[11] R. Xiao, J. Schwarz, N. Throm, A. D. Wilson, and H. Benko, "MRTouch: Adding touch input to head-mounted mixed reality," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 4, pp. 1653–1660, Apr. 2018.
[12] T. Han, "Exploring design factors for transforming passive vibration signals into smartwear interactions," in *Proc. 9th Nordic Conf. Hum.-Comput. Interaction*, 2016, pp. 1–10.
[13] J. McIntosh, A. Marzo, and M. Fraser, "Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection," in *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol.*, 2017, pp. 593–597.
[14] X. Jiang et al., "Exploration of force myography and surface electromyography in hand gesture classification," *Med. Eng. Phys.*, vol. 41, pp. 63–73, 2017.
[15] V. Becker et al., "Touchsense: Classifying finger touches and measuring their force with an electromyography armband," in *Proc. ACM Int. Symp. Wearable Comput.*, 2018, pp. 1–8.
[16] Z. Xiao and C. Menon, "A review of force myography research and development," *Sensors*, vol. 19, Oct. 2019, Art. no. 4557.
[17] D. Farina et al., "The extraction of neural strategies from the surface EMG," *J. Appl. Physiol.*, vol. 96, pp. 1486–1495, Apr. 2004.
[18] R. Ma, L. Zhang, G. Li, D. Jiang, S. Xu, and D. Chene, "Grasping force prediction based on sEMG signals," *Alexandria Eng. J.*, vol. 59, no. 3, pp. 1135–1147, 2020.
[19] C. Wu et al., "Optimal strategy of sEMG feature and measurement position for grasp force estimation," *PLoS one*, vol. 16, no. 3, 2021, Art. no. e0247883.
[20] X. Jiang, L.-K. Merhi, and C. Menon, "Force exertion affects grasp classification using force myography," *IEEE Trans. Hum.-Mach. Syst.*, vol. 48, no. 2, pp. 219–226, Apr. 2018.
[21] D. Farina et al., "The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 797–809, Jul. 2014.
[22] Y. Shi et al., "Ready, steady, touch!: Sensing physical contact with a finger-mounted IMU," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 4, pp. 1–25, Jun. 2020.
[23] Y. Gu et al., "Accurate and low-latency sensing of touch contact on any surface with finger-worn IMU sensor," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, 2019, pp. 1059–1070.
[24] J. de Moraes et al., "Advances in photopletysmography signal analysis for biomedical applications," *Sensors*, vol. 18, no. 6, 2018.
[25] M. Sikora and S. Paszkiel, "Muscle activity measurement using visible light and infrared," *IFAC-PapersOnLine*, vol. 52, no. 27, pp. 329–334, 2019.
[26] A. T. Maereg et al., "Hand gesture recognition based on near-infrared sensing wristband," in *Proc. 15th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2020, pp. 110–117.
[27] E. Nsugbe, C. Phillips, M. Fraser, and J. McIntosh, "Gesture recognition for transhumeral prosthesis control using EMG and NIR," *IET Cyber- Syst. Robot.*, vol. 2, no. 3, pp. 122–131, 2020.
[28] Z. G. Xiao and C. Menon, "Counting grasping action using force myography: An exploratory study with healthy individuals," *JMIR Rehabil. Assistive Technol.*, vol. 4, 2017, Art. no. e6901.
[29] Y. Zhang et al., "ActiTouch: Robust touch detection for on-skin AR/VR interfaces," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, 2019, pp. 1151–1159.
[30] A. Bashkatov, E. Genina, and V. Tuchin, "Optical properties of skin, subcutaneous, and muscle tissues: A review," *J. Innov. Opt. Health Sci.*, vol. 4, pp. 9–38, Jan. 2011.
[31] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 66–77, Feb. 2016.
[32] Microsoft, "Windows app development - touch interactions." Accessed: Feb. 24, 2022. [Online]. Available: https://docs.microsoft.com/en-us/windows/apps/design/input/touch-interactions
[33] Apple, "iPhone user guide - basic gestures." Accessed: Feb. 24, 2022. [Online]. Available: https://support.apple.com/guide/iphone/learn-basic-gestures-iph75e97af9b/ios
[34] "Success criterion 2.5.8: Pointer target spacing" Web Content Accessibility Guidelines (WCAG 2.2), Tech. Rep. 2.5.8, 2021.
[35] H. Liu et al., "High-fidelity grasping in virtual reality using a glove-based system," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 5180–5186.
[36] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[37] Y. Shevchuk et al., "NeuPy - neural networks in python." Accessed: Feb. 24, 2022. [Online]. Available: http://neupy.com/pages/documentation.html
[38] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems." Accessed: Feb. 24, 2022. [Online]. Available: https://www.tensorflow.org/
[39] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.