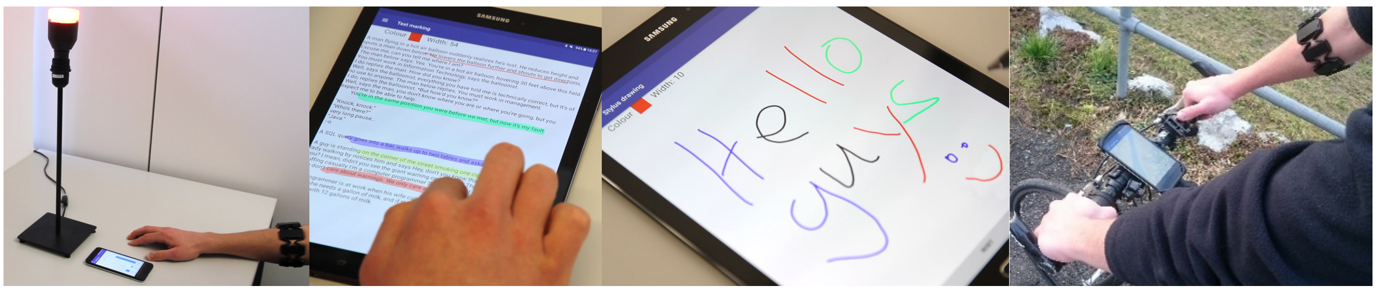
TouchSense: Classifying Finger Touches and Measuring their Force with an Electromyography Armband

Vincent Becker, Pietro Oldrati, Liliana Barrios, Gábor Sörös
Department of Computer Science, ETH Zurich, Switzerland
{vbecker | oldratip | lilianab | soeroesg}@ethz.ch



(a) (b) (c) (d)

Figure 1: We propose a method that allows to determine the finger and the force applied in touches. We present several applications using this method to enrich the interaction with devices.

ABSTRACT
Identifying the finger used for touching and measuring the force of the touch provides valuable information on manual interactions. This information can be inferred from electromyography (EMG) of the forearm, measuring the activation of the muscles controlling the hand and fingers. We present TouchSense, which classifies the finger touches using a novel neural network architecture and estimates their force on a smartphone in real time based on data recorded from the sensors of an inexpensive and wireless EMG armband. Using data collected from 18 participants with force ground truth, we evaluate our system's performance and limitations. Our system could allow for new interaction paradigms with appliances and objects, which we exemplarily showcase in four applications.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces

Author Keywords
Finger Identification; Wearable Computing; Touch-based interfaces; EMG; Interaction; CNN; LSTM

ISWC '18, October 8–12, 2018, Singapore, Singapore

DOI: https://doi.org/10.1145/3267242.3267250

INTRODUCTION
Finger touches, presses, and grasps are among the most important ways humans interact with their surroundings. Evolutionary biologists speculate that the ability to manipulate objects is to a large extent responsible for the remarkable development of primates' brains, which ultimately made humans the dominant species of the planet [9]. Tools built throughout human history take advantage of the wide spectrum of human touch. A piano, for example, can only be meaningfully played with several fingers at once, and its keys react to the force applied with differently loud tones. This richness of human touch, however, is only poorly represented in most technological devices. A traditional light switch, for example, is insensitive to both the pressure of the touch, and to the finger(s) used in the process. It will uniformly turn on the light, with the same colour and brightness. Taking the pressure of the touch gesture into account already yields more sophisticated devices, such as dimming light switches. Considering also the specific finger used in the interaction can add new and more features, and perhaps open entirely new interaction possibilities.

The lack of adaptation leaves space to explore the use of techniques that identify the finger and estimate the force (or pressure) of finger touches. Nevertheless, even amongst the newer touch-based devices such as tablets and smartphones only very few feature a pressure-sensitive input surface which do not have the means of identifying the finger either. Apple's newer products incorporate the technology "Force Touch"[1], which is able to distinguish between different degrees of force being applied to the screen, thereby adding an input dimension.

[1] https://en.wikipedia.org/wiki/Force_Touch

Instead of augmenting specific devices with such sensing capabilities, it is beneficial to augment the humans themselves to sense the fingers used and the corresponding force of their touches. That would allow any surface on any object in the environment to act as an input interface and enrich the human's interaction capabilities. For example, one could turn on the lamp by pressing on a surface nearby, such as a wall or a table, or control the TV from the sofa by touching the armrest. For such a system augmenting the human to work, one would either have to augment the user's fingertips with force sensors, which is rather obtrusive, or measure the contraction of the muscles in the forearm to infer the finger used and estimate the force exerted by that finger, since the muscles in the forearm control the fingers. The latter is possible using electromyography (EMG), the measurement of muscle activation potentials. EMG devices are usually very expensive and require large amplifiers; however, in recent years, an inexpensive and wireless EMG armband, the Thalmic Labs Myo[2], has become commercially available.

In this paper, we present *TouchSense*, a system for classifying finger touches of the thumb, forefinger, and middle finger and estimating the exerted force. We only utilize the Thalmic Lab's Myo EMG armband and a standard Android smartphone which continuously receives the EMG signal via Bluetooth, making our system wireless and mobile. For finger identification, we use a light-weight neural network running on the smartphone with a short average inference time of under 10 ms, thus fulfilling real time requirements. Furthermore, the network has a model size of less than 140 KB. Subsequently, the force is estimated and output on a continuous scale. We collected the necessary data from 18 participants and measured force ground truth values with a self-built hardware setup. This enables us to train a regressor for the true force values. We evaluate our system in several experiment designs, including a user-independent design, a user-dependent but session-independent design, and a session-dependent design. Moreover, we demonstrate its use in four demo applications. Overall results show that a user-independent system is difficult to achieve, however, a user-dependent, but session-independent system is possible, i.e. our system has to be trained once for every user.

BACKGROUND AND RELATED WORK

Electromyography (EMG) measures the electrical activation of a muscle. Muscle fibres are controlled by motor neurons by electrical impulses. These activations induce a measurable difference in the potential of the muscle cells. Thereby, each muscle contraction can be associated with an EMG signal. We use the Myo armband, a EMG device applied to the forearm, thereby measuring the muscles controlling hand and fingers. A great challenge when performing EMG analysis are the strong variations between people due to interpersonal differences in anatomical properties, such as muscle strength, position of bones, and skin conductance. Furthermore, the measurements of two sessions for the same person may differ because the exact placement of the electrodes changes.

Most previous work using EMG in the HCI domain has been done in the area of hand and finger gesture recognition [1, 3, 6, 10, 14, 18, 20, 23, 25]. However, the mentioned works investigate only full-hand gestures or coarse-grained finger gestures. Moreover, many previous works differ from ours by either using mostly expensive, specialized, wired or custom-built hardware for measurement, some with a high number of channels, performing an offline analysis, and not being able to run in real time, or combining EMG with other sensors.

There is relatively little work considering finger touches and the applied force. In the following, we briefly discuss previous research concerning finger touching gestures and the estimation of applied force. DiDomenico et al. show the feasibility of using EMG for finger strength regression for the purpose of evaluating the ergonomics of finger-intensive tasks [5]. They collect force ground truth in an experiment including 30 participants performing finger gestures (e.g. pinches) simulating hand-intensive tasks. EMG data was gathered using three wired electrodes attached to the forearm. For regression, they fit linear models and show that this results in an acceptable error. However, they do not use the models on unseen data in a test case and only perform an offline analysis.

Saponas et al. classify finger gestures engaging the fore- and middle finger, extended or curled, tap, and lift and also classify finger strength into hard and light [21]. They show that it is possible to determine the finger which is used for pressing and distinguish the two pressure classes with high accuracy. However, they trained and tested with data from the same session. When performing a cross-user-validation, the performance decreases significantly. While they only classify into two force levels, we envision a system which estimates a continuous force level. Their EMG electrodes are wired to an expensive measurement setup with a high sampling rate of 2048 Hz (more than ten times higher than for our device). Moreover, they only perform an offline analysis. They continued their work and also built an online system for pinching gestures [22] using a custom-built wireless EMG armband. They show that it is possible to obtain over 70% accuracy in a two-session experiment design for pinching gestures using the fore-, middle, and ring finger. We go further and perform the analysis directly on a smartphone, thereby creating a completely mobile system using off-the-shelf hardware.

Benko et al. have built on the work done in [21] to perform finger identification and additionally estimate the finger pressure by using a smoothed average over all EMG channels [2]. They use this pressure estimate in a combination with the touch-sensing capabilities of a tabletop computer to allow extended input, such as adapting the width of a stroke in a painting application according to pressure. However, as they use the same system as in [21], their setup is also static and processed on a desktop machine. Furthermore, their design is session-dependent (i.e. it needs to be trained for each participant before every use and not only once per participant) and as they never collected force ground truth, it is not possible to say how well their estimate correlates to the actual force exerted by the finger onto the screen. In contrast, we collect ground truth data using our hardware setup and fit a regressor to the EMG data and the ground truth.

[2] https://www.myo.com/

Various other modalities have been examined for achieving finger identification. Vision-based approaches employ cameras viewing the scene from above tracking the users' fingers [4, 15, 19, 26]. Others view transparent displays from below and recognize individual fingerprints [13]. While these approaches have the benefit of not having to augment the users with any sensors, the interaction is limited to a fixed space. Furthermore, it is not possible to determine the applied force.

Contrarily, other researchers attached sensors directly to the fingers or integrate them in a glove in order to identify which finger is used in an interaction [11, 12, 16, 17]. This allows highly accurate finger identification, however, none of the researchers show how to estimate the force of touch interactions. Most importantly, they are rather obtrusive systems in terms of finger interaction. Another solution is to incorporate RFID tags into fake finger nails, which are detected by RFID readers in the interaction devices [24]. While this provides a method which does not interfere with the finger interaction, it requires the interaction surfaces to contain an RFID reader. Besides, force estimation is not possible.

Concerning the measurement of force, manufacturers of commercial products focus on hardware solutions in touch screens, as the aforementioned ForceTouch from Apple, or touch pads, such as Synaptics' Force[3] pad, a laptop touch pad which is able to measure the force applied by the fingers. However, the input surface is confined to the relatively small size of the touch screen or touch pad, whereas our system can turn any surface or object into an input surface.

In conclusion, the main differences to previous work are the following: we use an inexpensive, wireless, off-the-shelf EMG armband with a low sampling rate for finger classification and force estimation; we collect force-ground truth for finger presses and are thereby able to evaluate our force estimation and finally we run an online, real time analysis on a smartphone, through which our system becomes mobile.

THE TOUCHSENSE SYSTEM

Data Collection

To record EMG data, we use the Myo armband from which we deliver the EMG data to an Android smartphone via Bluetooth. Additionally, we collect ground truth force data using a measurement setup. This allows us to train regressors for the exerted strength and the force values also serve as labels for classification (then reduced to zeros and ones). The setup consists of a measurement circuit using three movable force-sensitive resistors wired to an Arduino Yún, which is itself connected to a computer. The force-sensitive resistors are fixed to a board by Velcro pads, in order to adjust them to each participant's hand anatomy. We provide a circuit diagram in the supplementary material for potential follow-up research. The smartphone and the Arduino are synchronized via an NTP time server. Note that the setup for ground truth is only required for collecting training data and not at test time or in a real application. We convert the force signal to Newtons (resolution 0.01 N). Typical pressing forces that can be comfortably exerted range from 0 to 12 N. The Myo samples at a rate of 200 Hz from eight channels (i.e. eight sensors) in a value range from -128 to 127 (unitless). The Myo's sampling rate is relatively low when compared to standard EMG measurement devices (sampling rate commonly around 2 kHz). Potential line noise interference at the frequencies 50 Hz and 60 Hz are automatically filtered.

[3] https://www.synaptics.com/products/touchpad-family/forcepad

Each session consists of a series of presses for each finger, with soft (around 2 N to 4 N) and strong presses (around 8 N to 12 N), and a period where the participant modulates the pressing force from low to high (i.e. from 0 N to 12 N). This is done for the thumb, the forefinger, the middle finger, and also all combinations of the three. Because we also record combinations, we only included three digits in the data collection in order to keep sessions short. A single session takes around five to seven minutes. We segment every recording in time windows with a length of 10 EMG samples with an overlap of 9, i.e. every window represents a time span of 50 ms over eight EMG channels and for every 5 ms of recording we produce a new window. This way we generate a high number of training samples even for short data collection trials.

We collected data from 18 participants (six females, 20 to 66 years old, average age 27 years). During data collection, the participants were sitting at a table. We tried to keep the placement between different sessions and participants as similar as possible. In total we gathered 1,819,846 samples. All our participants wore the armband on the right arm, however it should be simple to integrate also the left-handed case by mirroring the EMG channels [14]. For fifteen participants, we collect a second EMG session for a user-dependent, but session-independent experiment design. For two of these, we recorded another seven sessions, i.e. nine in total, in order to examine the effect of utilizing an increasing amount of training data. For all the participants with at least two sessions, the recordings were carried out at least one day apart from each other. As mentioned above, our dataset also includes touch and force data for any combinations of thumb, forefinger, and middle finger which were collected in the same sessions and the same way as the single touch gestures. They are not used in the evaluation of this paper, but we plan to do so in the future.

Finger Classification and Force Estimation

As dataset for the classification task, we use all samples with forces over 2 N for a single finger and under 1 N for the other two fingers to remove finger combinations, and converted the force values into classification labels. In most previous works based on EMG, researchers have used hand-crafted features for classifiers such as Support Vector Machines. In first experiments using the data of all participants, we evaluated several approaches with several different sets of features gathered from the literature. However, although we tested a vast set of combinations of features, we could not exceed a test accuracy of around 65% on our dataset. Thus, we decided to design a neural network for finger classification with the intention that the network would learn the best features, relieving us from the requirement of performing feature engineering and selection, an approach recently also followed by other researchers [7, 8].

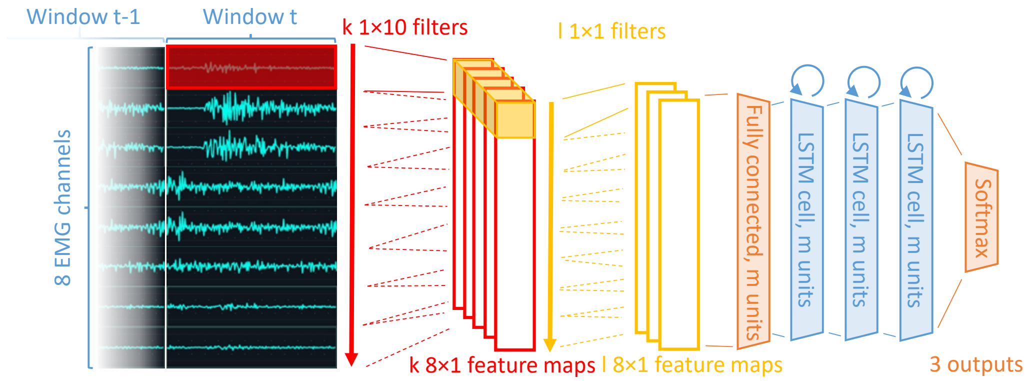


Figure 2: Our architecture for finger classification consisting of two convolutional layers with k and l filters, one fully-connected layer with m units, three stacked LSTM cells with m units each, and finally a softmax layer. The LSTM cells take five consecutive windows into account.

Our architecture is depicted in Figure 2. Since hand-crafted features are mostly computed per EMG channel, we decided to do the same. The first network layer learns features per channel. Since we want it to learn the same features for each channel, the weights of the corresponding neurons are shared. This has the beneficial property of being the same as a convolution over an image with $1 \times w$ filters without padding, where w is the width of the image. In our case the "image" consists of eight EMG channels with a window of 10 values each, hence the filters have the shape 1×10. The result of this first layer are k 8×1 feature maps, where k is the number of filters. Keeping k variable allows to control the complexity of the network. To reduce the number of feature maps we add a convolutional layer with l 1×1 filters (l should be smaller than k). These take all k features from the previous layer into account and produce l higher-level features.

The feature maps are then reshaped into a single vector which is the input to the rest of the network. It consists of a fully connected layer with m units, three fully connected LSTM (Long Short Term Memory) cells, also with m units each, and a softmax layer to output the probabilities for each finger. In the fully connected layer we use dropout with a dropping probability of 0.5. The fully connected layer is able to take every combination of features resulting from the convolutional layers into account. This is a great advantage compared to other classifiers, where these combinations of channel features have to be encoded in specific multi-channel features (c.f. for example [6]). The motivation for the LSTMs is that they allow us to take information over several windows into account, thus allowing to exploit both temporally local information from single windows as well as longer dependencies. We train the LSTM cells on sequences of five windows and during testing also feed sequences of five windows. The number of outputs of the softmax layer corresponds to the number of fingers taken into account in the specific experiment, i.e. there are either two or three. The number of feature maps in the convolutional layers, and the number of units in the LSTM cells are given as variables to control the complexity of the network to adapt to different amounts of available training data. We use a cross-entropy loss function, an Adam optimizer with a learning rate of 0.0025 and a batch size of 200. The number of training epochs varies for the different experiments, to control for overfitting. We implemented the network in TensorFlow[4], trained it on a computer and exported the model file to our Android application. The model size is only about 140 KB. The runtime for all our models is below 10 ms on average on an LG Nexus 5X and hence fulfils real time requirements for human-computer interaction. We smooth the predictions using a majority vote over the last five predictions to make the results more stable. In the evaluation this is only done when we test on whole sessions (either in the user-independent or the user-dependent, session-independent setting), as the smoothing requires time-continuous sequences.

To obtain a force value for the finger press, we use the mean absolute value over all channels and the whole window (MAV) as calculated by $MAV = \frac{1}{8*w}\sum_1^8\sum_1^w |v_{ij}|$ where i indexes one out of the eight channels and j indexes one out of the w samples per window. This is a good indicator of force (cf. Section "Evaluation"), which does not have to be learnt. As the predictions, we also smooth the MAV to make the values more stable. If we allow user-dependent calibration, we can also fit a regression function. We use a linear regression model solely based on the MAV to map the MAV to actual force values. The code and data are available at https://github.com/vincentbecker/TouchSense.

EVALUATION

Finger Classification

We carry out several experiments, including mixing all data and performing cross-validation, a user-independent, and a session-independent design. For all our experiments, the description, including the network configuration, the evaluation method, and the average accuracy are given in Table 1. Whenever applicable, we perform 30-fold cross-validation, otherwise we apply cross-validation across participants or multiple sessions of the same participant. For the user-independent (no. 2 and 5) and the session-independent experiments (no. 3 and 6) we apply prediction smoothing as mentioned in Section "Finger Classification and Force Estimation". For the

[4] www.tensorflow.org

Experiment no.	Setting	Cross-validated	Fingers	Configuration	Accuracy
1	All data mixed	30-fold	t, f, m	64, 8, 256	97.4%
2	User-independent	Across users	t, f, m	64, 8, 256	48.4%
3	User-dependent, session-independent	Across sessions per participant	t, f, m	32, 4, 32	72.6%
4	All data mixed	30-fold	t, m	64, 8, 256	98.7%
5	User-independent	Across users	t, m	64, 8, 256	70.1%
6	User-dependent, session-independent	Across sessions per participant	t, m	32, 4, 32	86.8%

Table 1: The description of the experiments including the cross-validation, which fingers were included (t: thumb, f: forefinger, m: middle finger), and the network configuration (*k, l, m*). The results are given as the average accuracy over all folds and participants.

other experiments, the data is shuffled, hence it loses its sequential character and smoothing is not applicable. We always balance the data for training, i.e. we use the same number of samples per class to avoid skewed data proportions in the training process. Whenever possible, we also give more detailed performance figures in the form of confusion plots and accuracy charts. The confusion plots display the normalized confusion matrix (all numbers divided by the total number of samples), which in case of experiments no. 2, 3, 5, and 6 is the average of the normalized confusion matrix for each participant (no. 2 and 5) or session (no. 3 and 6). The accuracy charts display the accuracy when testing on a single participant in the user-independent experiments (no. 2 and 5) or the average test accuracy per participant for the session-independent experiments (no. 3 and 6).

Classification with three fingers

First of all, we perform our evaluation on all the data, including the thumb, the forefinger, and the middle finger.

Experiment 1

In a first experiment, we evaluated the setting of mixing the samples of all users and performing cross-validation. Due to the large number of samples, we deploy a relatively complex network. We train for 100 epochs and obtain a test accuracy of 97.4%. The confusion plot in Figure 6a shows how few misclassification there are. This proves that our neural network is able to capture the characteristics of the EMG signals.

Experiment 2

In a second experiment, we attempted the most challenging setting, a cross-validation on users, i.e. excluding a single participant's data from the training set and testing on this data in order to evaluate how user-independent our system is. Unfortunately, we obtain poor results for most participants as shown in Figure 3, and confirmed by the confusion matrix (c.f. Figure 6c). This is a result of the inter-personal anatomical differences and that the sensors are placed a little differently for every participant. It shows how individual the EMG data of each participant is. The user-independent design is generally a challenge in previous literature as well. The performance is generally poor, however our main goal and measure is high performance in the session-independent experiments.

Experiment 3

As a consequence of experiment 2, we moved to a user-dependent, but session-independent design in the third experiment, which resembles a real-world scenario where the system has to be trained once per user. For this purpose, we recorded at least two sessions for fifteen participants. We perform cross-validation on the sessions, i.e. we test on each session, after having trained on all the others. Note that for this experiment, we have much less training data, so we reduced the training epochs to 20 and also the layer sizes as shown in Table 1 to avoid overfitting. The results (c.f. Figure 4 and confusion matrix 6e) are much improved from the user-independent setting, as the variance now is only caused

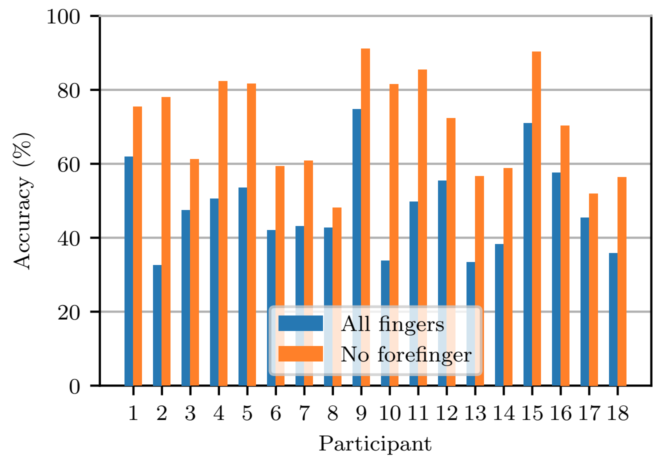


Figure 3: Accuracies for participants in experiments 2 and 5.

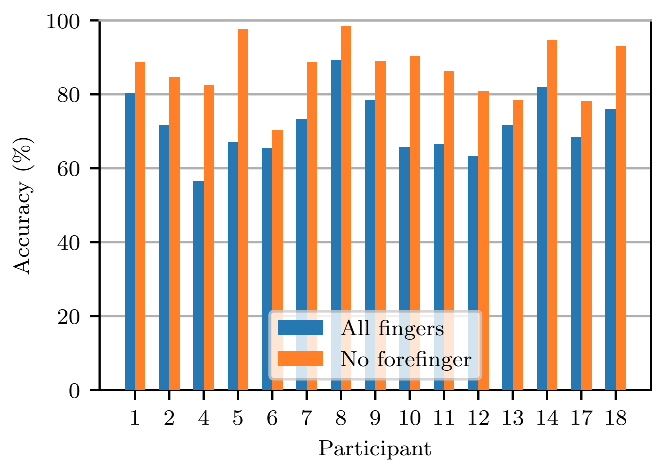


Figure 4: Accuracies for participants in experiments 3 and 6.

by differences between sessions, but not between people. As the sessions were at least a day apart, this proves that we can generalize over a longer time period. As our evaluation for the limited set of two participants with nine sessions each show (cf. Section "Finger Classification with more Data"), we can expect to obtain better results with more data per participant.

Classification with two fingers, experiments 4 to 6

A problem we identified and which is shown clearly by the confusion matrices in Figures 6a, 6c, and 6e is that the forefinger is often confused with either the thumb or the middle finger. We thus also investigated the case of only classifying thumb and middle finger and reran all our previous experiments. We show the result in the confusion matrices in Figures 6b, 6d, and 6f and as second bars in the accuracy figures. As expected, the results for this case are better than in the three-finger case for all experiments. Especially experiment 6 with over 86% shows that our system can be valuable for real-world applications.

Finger Classification with more Data

As mentioned above, for two participants we collected nine sessions to investigate the effect of training with more data on the classification accuracy in the session-independent case. In Figure 5 we show the accuracy when training on an increasing amount of sessions and performing cross-validation for each of the two participants (including all three fingers, i.e. resembling experiment 3). With more sessions used for training, the performance significantly increases for both participants. We believe that we could achieve a higher overall performance with more training data for each participant.

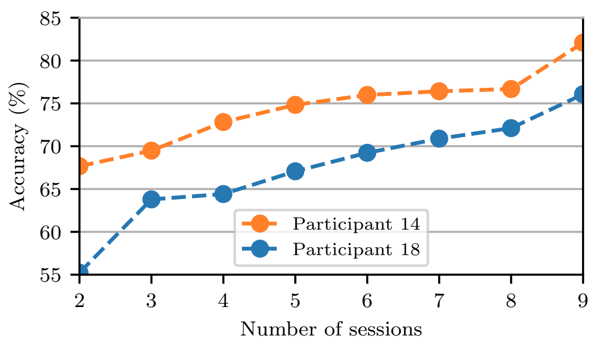


Figure 5: The accuracies in the session-independent experiment for two participants when increasing the number of sessions for training.

Force Estimation

As mentioned in Section "Finger Classification and Force Estimation" we use the mean absolute value over all channels and the whole window (MAV) as default force estimation. The Pearson's correlation coefficient of the force ground truth value and the MAV for all participants is 0.87 on average, which proves that the MAV is a good proxy for the force and changes in the MAV correspond to changes in the force. For participants with more than one session, we average the results over all of them.

Nevertheless, if data for calibration is available, e.g. in the multi-session setting, we can fit a regression function. For each participant, we fit a linear regression model based on the MAV and cross-validate it (10-fold). For participants with multiple sessions, we average their results again. The average mean absolute error (MAE) over all participants is 1.26 N, i.e. roughly 10% of the usual force range, which reaches up to 12 N. When smoothing the results over five predictions, as we do it also for the classification, the MAE decreases to 1.22 N. Figure 7 depicts the ground truth force values, the MAV (scaled by a factor of 50), and the predicted force values from a linear regression model for a segment of a session with 10,000 samples. It exemplifies that the MAV reflects changes in the force well. Nevertheless, the regression model performs better, especially in areas where no force is applied. Here the MAV is still positive, which results from sporadic EMG activation. For all participants with two sessions, we fit a linear regression function on one session and applied it on the other (and vice versa) without smoothing. The average MAE for this session-independent scenario is 1.28 N, which shows that the calibration of the regression system works across multiple sessions. The results of this evaluation are given in Figure 8.

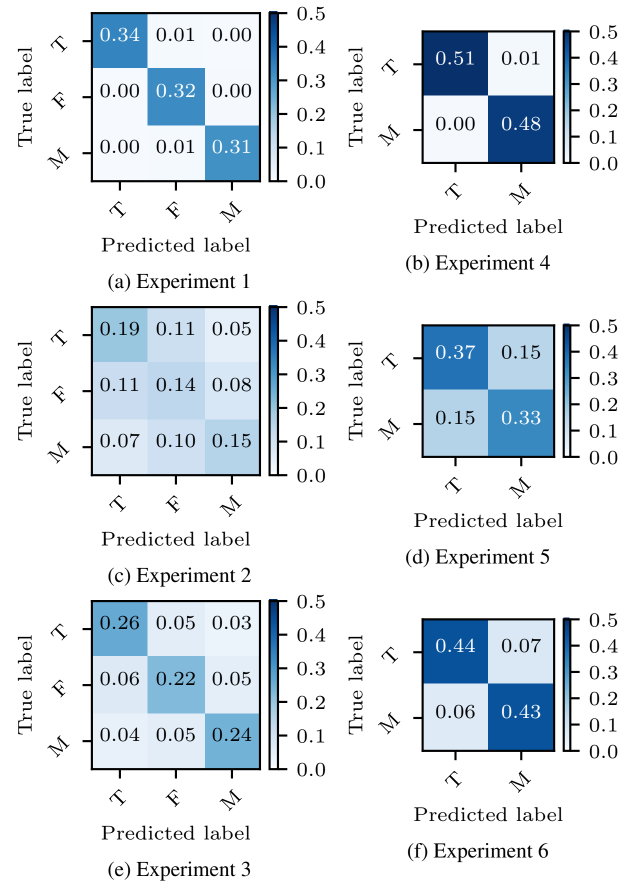


Figure 6: Confusion plots for all the experiments.

LIMITATIONS

The user-dependence still is a strong limitation, i.e. our system has to be trained per user. Besides, to be able to train the force regression, ground truth force values are necessary, which we

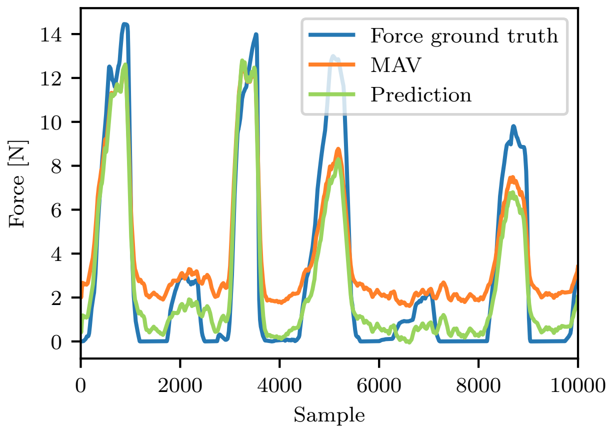


Figure 7: An example of the ground truth force values, the MAV, and the predictions of a linear regression model for 10,000 samples.

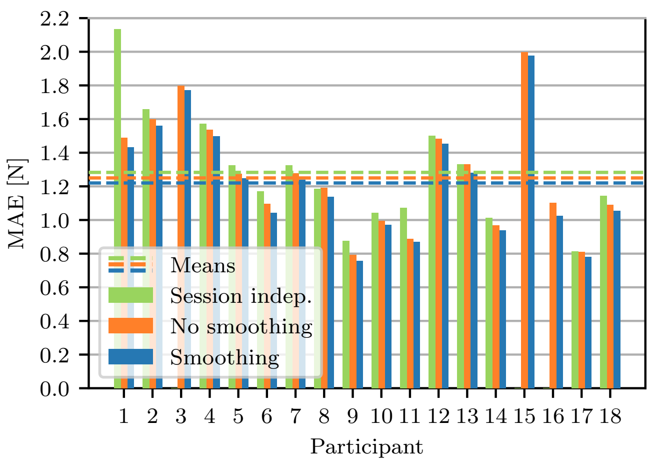


Figure 8: The mean absolute errors (MAE) when cross-validating linear regression models trained on the MAV and force ground truth from individual, and from multiple sessions.

measure with our hardware setup. However, this is not available everywhere and it does not appear reasonable for every user to have one. Nevertheless, the finger classification could be trained without the exact force information only by instructing the user to perform a series of finger presses and recording the data. This approach would not require any hardware apart form the smartphone and the Myo, which are anyway needed for the normal use. Furthermore, the user could still obtain a force estimate through the MAV which does not have to be learnt from training data. The user-dependence might be solved by using more data for training, as the original Myo gestures also work user-independently. A further practical limitation is that the Myo is not particularly comfortable to wear, especially after a longer period of time. However, we expect there to be lighter and more comfortable device developments in the future, maybe even sewn into clothing garments.

APPLICATIONS

We created several demo applications employing our system. They show that it is possible to use TouchSense in real-world scenarios. A visual overview of all the applications is shown in Figure 1. It is important to note that for all the demos we use the same model, i.e. did not finetune the model to the specific use case. Furthermore, in none of them the system was calibrated during the demo session. A video of our demos is included in the supplementary material.

The first application (Figure 1a) demonstrates how any surface can be used as an input surface in order to control smart devices without them being required to have an interaction surface of their own. Here, a smart lamp can be controlled by pressing on any surface around it. Pressing with the forefinger decreases the brightness, pressing with the middle finger increases it. The magnitude of the change adapts to the force applied by the finger, i.e. strong presses change the brightness quickly while soft presses change it slowly.

As second application (Figure 1b), we created a tablet app for marking text. On the screen any finger can be used for marking the text in a semi-transparent colour. We extend the interaction surface of the tablet by an imaginary colour palette controlled by finger presses. Pressing outside of the tablet screen, e.g. on a table, controls the palette. It includes three options: Pressing with the thumb changes the colour to the next colour, pressing with the forefinger sets the stroke width according to the strength of the touch, and pressing with the middle finger reverts the last stroke. By employing this concept we can mitigate the size limitations of the display.

We realized that our system also works for objects held in the hand without further training. We took advantage of this and created a tablet drawing application using a stylus (Figure 1c). The drawing behaviour adapts to the way the stylus is pressed with the fingers and provides a new drawing interface. When pressing with the thumb on the stylus, the colour is changed to the next colour, as before. Pressing strongly with all fingers activates an erase function. This shows that TouchSense cannot only enhance the interaction with smart devices such as tablets, but also extend the functionality of usual "non-intelligent" objects such as the stylus.

In our last demo we present an outdoor use case (Figure 1d). We create an Android map application for cycling, showing a map and the current position, which can be controlled by pressing on the bicycle's handlebar without letting it go, strongly facilitating the control of the map. While riding the bicycle, the cyclist can change the map type by pressing the thumb on the handlebar, zoom out by pulling on his / her forefinger, and zoom in by pulling on his / her middle finger.

CONCLUSION

We presented TouchSense, a mobile system to augment a human with a sensing mechanism to classify touch gestures including the thumb, the forefinger, and the middle finger and estimating the force of the touch from EMG data. Through TouchSense, all surfaces in the environment are turned into interaction areas. Besides, existing interaction paradigms can be enriched with additional features.

To gather the EMG data, we employ an inexpensive, wireless EMG armband. The EMG data is then processed on a smartphone in real time to classify the finger used in the interaction and estimate the force applied. Moreover, for training we collect ground truth force data in order to provide labels and enable us to fit regression models for the force estimation. The classification is done by a convolutional neural network we specifically designed for the purpose of EMG analysis. It runs on a standard smartphone in under 10 ms per inference and is at most 140 KB large. The evaluation showed that, as expected also from previous works, a user-independent setting is challenging and the results are insufficient for a real application. Nevertheless, in a user-dependent, but session-independent setting our network produces satisfying results, especially when being trained and tested for only two fingers, which could be sufficient for many applications. We furthermore showcased several demos, which prove our system's real world value.

In the future we plan to collect more data in order to tackle the challenging user-independent scenario. Furthermore, we aim at improving our classification neural network and in particular want to design more sophisticated force regression methods, such as a regression network running in parallel to the classification or a unified architecture for both tasks. Finally, another challenge we consider for the future is classifying not only single fingers, but also finger combinations.

REFERENCES

1. C. Amma, T. Krings, J. Böer, and T. Schultz. Advancing Muscle-Computer Interfaces with High-Density Electromyography. In *Proc. CHI '15*.
2. H. Benko, T. Saponas, D. Morris, and D. Tan. Enhancing Input on and Above the Interactive Surface with Muscle Sensing. In *Proc. ITS '09*.
3. X. Chen, X. Zhang, Z. Zhao, J. Yang, V. Lantz, and K. Q. Wang. Hand Gesture Recognition Research Based on Surface EMG Sensors and 2D-accelerometers. In *Proc. ISWC '07*.
4. A. Colley and J. Häkkilä. Exploring Finger Specific Touch Screen Interaction for Mobile Phone User Interfaces. In *Proc. OzCHI '14*.
5. A. DiDomenico and M. Nussbaum. 2008. Estimation of forces exerted by the fingers using standardised surface electromyography from the forearm. *Ergonomics* 51, 6 (2008).
6. I. Donovan, J. Puchin, K. Okada, and X. Zhang. Simple Space-Domain Features for Low-Resolution sEMG Pattern Recognition. In *Proc. EMBC '17*.
7. Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng. 2017. Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation. *Sensors* 17, 3 (2017).
8. Y. Du, Y. Wong, W. Jin, W. Wei, Y. Hu, M. Kankanhalli, and W. Geng. Semi-supervised Learning for Surface EMG-based Gesture Recognition. In *Proc. IJCAI '17*.
9. P. Ehrlich and A. Ehrlich. 2008. *The Dominant Animal*.
10. K. Englehart and B. Hudgins. 2003. A Robust, Real-time Control Scheme for Multifunction Myoelectric Control. *IEEE Trans. Biomedical Engineering* 50, 7 (2003).
11. M. Fukumoto and Y. Suenaga. "FingeRing": A Full-time Wearable Interface. In *Proc. CHI '94*.
12. A. Gupta and R. Balakrishnan. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proc. CHI '16*.
13. C. Holz and P. Baudisch. Fiberio: A Touchscreen that Senses Fingerprints. In *Proc. UIST '13*.
14. F. Kerber, M. Puhl, and A. Krüger. User-independent Real-time Hand Gesture Recognition Based on Surface Electromyography. In *Proc. MobileHCI '17*.
15. S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with Large Displays from a Distance with Vision-tracked Multi-finger Gestural Input. In *Proc. UIST '05*.
16. N. Marquardt, J. Kiemer, D. Ledo, S. Boring, and S. Greenberg. Designing User-, Hand-, and Handpart-aware Tabletop Interactions with the TouchID Toolkit. In *Proc. ITS '11*.
17. D. Masson, A. Goguey, S. Malacria, and G. Casiez. WhichFingers: Identifying Fingers on Touch Surfaces and Keyboards Using Vibration Sensors. In *Proc. UIST '17*.
18. J. McIntosh, C. McNeill, M. Fraser, F. Kerber, M. Löchtefeld, and A. Krüger. EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. In *Proc. CHI '16*.
19. S. Murugappan, Vinayak, N. Elmqvist, and K. Ramani. Extended Multitouch: Recovering Touch Posture and Differentiating Users Using a Depth Camera. In *Proc. UIST '12*.
20. A. Samadani and D. Kulic. Hand Gesture Recognition based on Surface Electromyography. In *Proc. EMBC '14*.
21. T. Saponas, D. Tan, D. Morris, and R. Balakrishnan. Demonstrating the Feasibility of Using Forearm Electromyography for Muscle-computer Interfaces. In *Proc. CHI '08*.
22. T. Saponas, D. Tan, D. Morris, J. Turner, and J. Landay. Making Muscle-computer Interfaces More Practical. In *Proc. CHI '10*.
23. E. Scheme and K. Englehart. On the Robustness of EMG Features for Pattern Recognition based Myoelectric Control; A Multi-dataset Comparison. In *Proc. EMBC '14*.
24. K. Vega and H. Fuks. Beauty Tech Nails: Interactive Technology at Your Fingertips. In *Proc. TEI '14*.
25. X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang. 2011. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Trans. on Systems, Man, and Cybernetics* 41, 6 (2011).
26. J. Zheng and D. Vogel. 2016. Finger-Aware Shortcuts. In *Proc. CHI '16*.