ThumbPitch: Enriching Thumb Interaction on Mobile Touchscreens using Deep Learning

Jamie Ullerich
University of Stuttgart
Stuttgart, Germany
st142792@stud.uni-stuttgart.de

Maximiliane Windl
LMU Munich
Munich, Germany
Munich Center for Machine Learning
Munich, Germany
maximiliane.windl@ifi.lmu.de

Andreas Bulling
University of Stuttgart
Stuttgart, Germany
andreas.bulling@vis.uni-stuttgart.de

Sven Mayer
LMU Munich
Munich, Germany
info@sven-mayer.com

ABSTRACT

Today touchscreens are one of the most common input devices for everyday ubiquitous interaction. Yet, capacitive touchscreens are limited in expressiveness; thus, a large body of work has focused on extending the input capabilities of touchscreens. One promising approach is to use index finger orientation; however, this requires a two-handed interaction and poses ergonomic constraints. We propose using the thumb's pitch as an additional input dimension to counteract these limitations, enabling one-handed interaction scenarios. Our deep convolutional neural network detecting the thumb's pitch is trained on more than 230,000 ground truth images recorded using a motion tracking system. We highlight the potential of ThumbPitch by proposing several use cases that exploit the higher expressiveness, especially for one-handed scenarios. We tested three use cases in a validation study and validated our model. Our model achieved a mean error of only 11.9°.

CCS CONCEPTS

• **Human-centered computing** → **Touch screens**; Empirical studies in HCI; • **Hardware** → *Touch screens.*

KEYWORDS

ThumbPitch, thumb, one-handed, interaction technique, input, deep neural network, convolutional neural network

ACM Reference Format:
Jamie Ullerich, Maximiliane Windl, Andreas Bulling, and Sven Mayer. 2022. ThumbPitch: Enriching Thumb Interaction on Mobile Touchscreens using Deep Learning. In *Proceedings of the 34th Australian Conference on Human-Computer Interaction (OzCHI '22), November 29-December 2, 2022, Canberra, ACT, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3572921.3572925

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
OzCHI '22, November 29-December 2, 2022, Canberra, ACT, Australia
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0024-8/22/11…$15.00
https://doi.org/10.1145/3572921.3572925

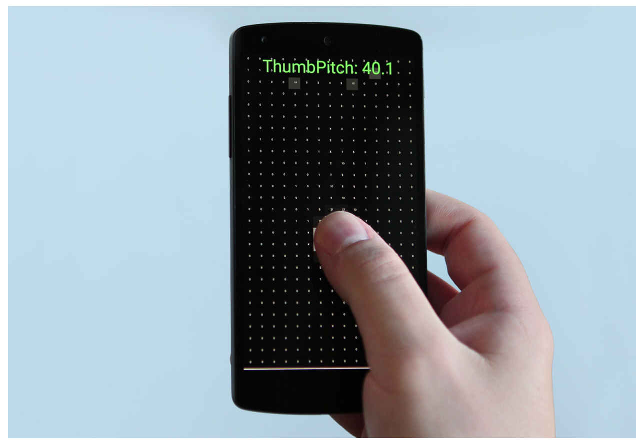


Figure 1: A developer mode of *ThumbPitch* input, with the capacitive image on the screen and the correlated pitch value of the thumb.

1 INTRODUCTION

Touch-based devices dominate the interaction landscape and can be found not only in mobile devices, such as smartwatches, smartphones, and laptops, but also in devices traditionally with haptic buttons, such as door-lock pads or household appliances. Touchscreens are attractive because they combine input and output in a single interface. Thus, they are easy to learn and use. However, the simplicity of touch interaction comes with several limitations, the most prominent being limited expressiveness; that is, today's touch controllers only extract a 2D point from the finger touching the surface. Yet, finger input contains much more information, such as finger type, pressure, and orientation.

Consequently, an increasing number of researchers and manufacturers have investigated ways to extend the capabilities of touchscreens with additional input dimensions. For instance, *Knuckle-Touch* [13, 34], an enriching interaction that nowadays can be found in a wide variety of Huawei phones. However, this approach or others like Apple's ForceTouch are still limited in expressiveness as they can only input two or three levels (finger vs. knuckle). Beyond enriching interactions that made it into consumer products, a wide variety of interactions for touch-based devices has been proposed over the last decades, such as finger-identification [10, 19, 24], finger-authentication [11, 17], the finger roll [32], and most prominently, finger orientation [27, 31, 36, 39]. Finger orientation offers the largest input space with two dimensions; however, the enlarged input comes at an ergonomic cost [26, 28]. Boring et al. [3] present a promising approach by approximating the thumb contact area as additional input. However, they only rely on a single value to estimate the input: the ellipse major. Due to the lack of fine-grained touch information, this implementation requires a one-time setup process. Moreover, the user has to specify their input hand for every interaction. Finally, their approach assumes that the change in ellipse major is linearly correlated with the pitch. Their work hinted that the contact area or the shape could improve their accuracy. We follow up on the hints by Boring et al. [3] and improve their estimation approach using deep learning.

We present *ThumbPitch*: a deep learning approach to estimate the thumbs' pitch angle to allow for a richer interaction, see Figure 1. At its core, *ThumbPitch* uses a convolutional neural network (CNN) regression model to estimate the pitch of the thumb based on the raw capacitive image recorded by the touch controller. In detail, we trained our model on data collected in a ground truth data collection study. Finally, we present several use cases that enrich touch interactions using *ThumbPitch*, of which we evaluate three in a separate study. We demonstrate that the method is accurate and robust across users when trained on a suitable amount of data.

ThumbPitch has two key advantages over existing touch input techniques. First, the nature of *ThumbPitch* allows for a one-handed interaction, which makes it the only technique that enables one-handed scenarios to support more than three input levels allowing for richer interaction. Second, *ThumbPitch* is highly practical, given that our deep learning estimation model can be deployed with a simple software update. This makes our technique directly applicable to the billions of touch screens already deployed worldwide. In summary, we contribute to the design, implementation, and evaluation of our *ThumbPitch* deep learning approach. A one-handed interaction that enables continuous value input. Moreover, we open-source *ThumbPitch*, allowing others to improve and deploy it.

2 RELATED WORK

While the literature on extending the interaction space for mobile devices and touch-based devices cover a wide range of possibilities, we will focus on techniques that directly impact or inspire *ThumbPitch*. Boring et al. [3] first proposed the idea of using the thumb as an additional input dimension. However, in their early implementation, they only allowed for a set of levels as input. Thus, continuous input was not possible. We will focus on ergonomic constraints during input and then on interaction techniques that only use the front screen to understand this limitation.

2.1 Reachability and Ergonomic Constraints

Mobile devices are distinctly different from stationary systems, such as PCs, in terms of affordances and ergonomic constraints, as the dominant input is touch. While today's touch controllers only extract a 2D point, Holz and Baudisch [15, 16] showed that touch input is multi-dimensional. Nevertheless, grip and reachability impact the usability of touchscreens. Bergstrom-Lehtovirta and Oulasvirta [1] studied the thumb's reachability for smartphones. They showed a correlation between surface size, hand size, and position of the index finger. Also, the finger orientation input has been shown to have heavy ergonomic constraints when using the index finger in a two-handed scenario [28]. At the same time, Wolf et al. [37] showed a high fidelity of the thumb during interaction with tablets. Further, Trudeau et al. [35] argued that buttons close to the resting position of the thumb are optimal for frequently used actions, while all other positions need more effort to be reached. Yet, Le et al. [25] showed that the thumb could reach large portions of the front screen without changing the grip. Thus, we argue that *ThumbPitch* has great potential to overcome current drawbacks.

2.2 Extended Interaction for Mobile Devices

As the contact area is often provided already up to the application layer in today's operating systems, one of the simplest enrichments for touchscreens is using the fingers' contact area [3, 8]. Similarly, Roudaut et al. [32] proposed rotating the finger around the roll vector by classifying left and right rolls based on the touchpoints. Other enrichments use separate sensors or data that are not provided to the application layer. One prominent example is detecting different parts of the finger touching the screen, which was initially envisioned by Harrison et al. [13]. They used the sound emitted by fingers when touching the surface to recognize the finger part. Later, Schweigert et al. [34] presented a deep neural network approach to use the capacitive image to distinguish between finger and knuckle touches. Initial work by Colley and Häkkilä [6] proposed finger-aware interaction. However, they used the Leap Motion to prototype the interaction, which made the device bulky. Therefore, Zhang et al. [40] used the built-in capacitive image coupled with electric field sensors to enhance pen and touch interactions using only small external sensors. Finally, Le et al. [24] presented a deep CNN recognizer to enable finger identification on a commodity smartphone without any external hardware. As a result of this development, we argue that using the capacitive image from the touchscreen has a high potential to enrich mobile interactions. Moreover, as today's mobile devices all use capacitive sensors, deploying new interaction techniques, such as *ThumbPitch*, is possible with only a simple software update.

Capacitive images have already been proposed for a wide variety of new interactions on smartphones and smartwatches, such as authentication using fingers and other body parts [9, 11, 17, 30]. Especially, capacitive images coupled with machine learning (ML) show great potential and continuously outperform the baseline. For simple touch prediction, Kumar et al. [21] improved the touch accuracy by 23% over the baseline using capacitive images and CNN. Others enabled force touch input without additional sensors using neural networks [2]. In a similar fashion, Le et al. [22] used the capacitive image to enable palm touch input. Another approach presented by Cami et al. [5] used the capacitive image and s on touch tables to enhance stylus input. They used different hand poses to activate a range of input modifiers, such as handwritten input and marking text. Finally, several researchers investigated the recognition of finger orientation [20, 27, 31, 36, 38, 39]. Remarkably, the most accurate implementation by Xiao et al. [38] improved the recognition by using only the built-in sensor of current mobile devices. Lastly, Mayer et al. [27] further improved these results using a deep CNN to estimate the fingers' pitch and yaw angles.

3 GROUND TRUTH DATA COLLECTION

We conducted a data collection study to train our deep CNN model to estimate the thumb's pitch angle. In line with previous work that also used raw capacitive images [2, 21], we aimed to collect a wide range of samples. Therefore, we collected samples with sub-millimeter accuracy at a high frame rate from 16 participants.

For a maximum variety of *ThumbPitch* inputs, we asked participants to *tap* with different angles on the screen and *continuously* change the angle without releasing the thumb. We asked participants to perform input at two different locations; in the *center* and at an *intersection* of the very center capacitive pixel of the capacitive image, following instructions by [27, 29]. Thus, we recorded four input modalities (tap vs. continuous × center vs. intersection). The order was randomized; however, we did not further consider the modalities in the evaluation as they only serve to capture a wide input variety, allowing for better generalizability.

3.1 Apparatus

We used an LG Nexus 5 (screen size: 4.95 in) mobile phone to collect the capacitive images as it provides us with the possibility to read the capacitive image in real time [23, 28]. In line with prior work [27, 33], we recorded the ground truth positions and angles of the finger and phone with a high-resolution motion capture system by OptiTrack.

We used Flex 3 cameras delivering high-precision marker positions at 100 FPS. We attached a 3D-printed marker on the right thumbnail to track the thumb angle and position. In detail, we used a specially designed 3D-printed marker attached to the participant's thumb to track the pitch angle while interacting with the phone. The 3D-printed marker represents the negative of a fingernail and fits on top of the participant's thumbnail without restricting reachability (see Figure 2). We further added three cylinders attached to infrared reflective markers for optical tracking.

We used a PC for data recording, live visualization, and synchronization, see Figure 2. To retrieve the capacitive images from the touchscreen, we modified the phone's kernel as described by Le et al. [23]. Therefore, we had access to the 27×15 8-bit raw capacitive images of the Synaptics ClearPad 3350 touch sensor with a frame rate of 20 FPS. Additionally, we implemented a custom app that instructed participants to repeatedly touch a red crosshair (size: 2×2 cm) to indicate the two positions (center vs. intersection).

3.2 Procedure

We welcomed participants into our laboratory and explained the data collection study to them in detail. After answering any questions they had, we asked them to sign an informed consent form. After asking participants to take a seat in front of our motion capture system, they filled in a demographics questionnaire. Using body tape, the experimenter then attached the marker to the participant's thumb; see Figure 2.

We asked participants to perform various pitch angles for the rest of the data collection study to record ground-truth pitch angles via the motion capture system and the raw capacitive images. During the recording, participants were asked to hold the phone with their dominant hand. Whenever participants had difficulties performing inputs, they were allowed to stabilize the phone with their non-dominant hand.

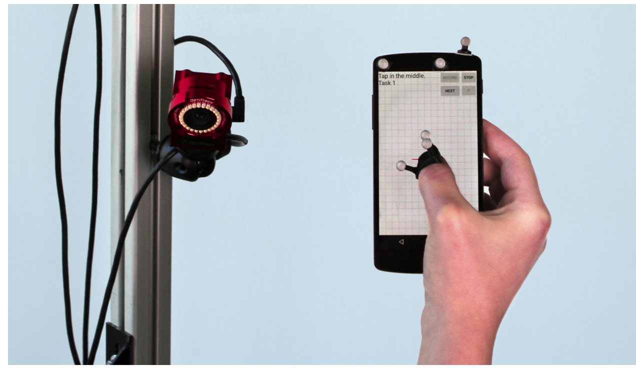


Figure 2: The setup of the data collection study, including the study phone, the optical reflection markers for tracking, and the OptiTrack cameras used for position tracking. The reflection markers are attached to the phone as well as the thumbs nail for submillimeter-accurate tracking.

During the experiment, the experimenter could monitor the participants' input progress using the live motion capture data. Thus, we could inform participants about the pitch angles they have not performed yet. This further enabled us to record a diverse dataset. We asked all participants to perform inputs in random order (tap vs. continuous × center vs. intersection).

3.3 Participants

We recruited 16 participants (6 females and 10 males) from an internal university volunteer pool aged between 18 and 32 years ($M = 24.5$, $SD = 3.5$). All participants' dominant hand was the right hand. No participant had any movement impairments. We paid 10 EUR per hour as compensation for the study, which latest approximately 45 minutes.

4 THUMBPITCH ESTIMATION MODEL

In the following, we present the development of the *ThumbPitch* estimation model. The resulting CNN regression model takes the raw capacitive images as input and estimates the pitch angle in angular degrees between 0° (flat finger) and 90° (steep finger).

4.1 Dataset and Preprocessing

While we automatically synchronized the capacitive images and ground truth pitch, the motion capture system has a latency of around 100 ms. Thus, we manually adjusted the synchronization using visual inspection of the marker movement and change in the pixel sum of the capacitive image. Afterward, we performed a blob detection similar to related work using capacitive images (e.g., [24, 27, 34]). Next, we cropped the blobs by the center of mass and pasted them in the upper left corner in an empty 14×14 image to counteract CNN translation effects [18]. We chose 14×14 as it is symmetric and the maximum blob plus overhead fits into the new size. While a larger image would be possible, no variation in

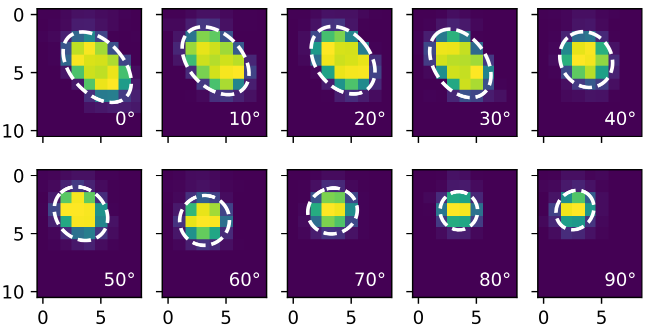


Figure 3: Ten example touches of one participant with the white dashed lines representing the ellipse fit.

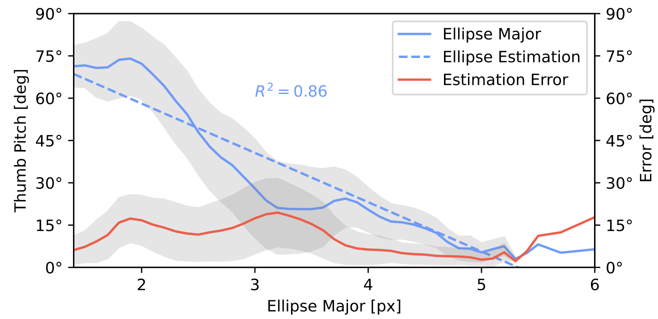


Figure 4: The correlation between the fitted ellipses major and the thumbs' ground truth pitch. The grey area represents the standard deviation.

data occurs in this area, thus, only making it bigger and slower. Examples of the processed images are shown in Figure 3.

After preprocessing, 233,072 capacitive images from 16 participants were ready to train our model. For each capacitive image, we have the corresponding ground truth pitch angle from the motion tracking system. Various impacting factors made the resulting dataset unbalanced, such as recording at 20 FPS, unfamiliarity with performing the extreme pitch movements [27, 38], and stability issues [3]. Therefore, we augmented the data by adding Gaussian distributed noise ($M = 4$, $SD = 8$) to balance all samples within the training dataset. We did not add noise to the test or validation set to avoid overfitting toward the noise but balanced the samples in the training dataset.

4.2 Boring et al.'s [3] Baseline Approach

Boring et al. [3] proposed that the contact size of the thumb can be used to identify the angle of the finger. In detail, they used the major radius of an ellipse fitting provided by the iPhone's API. Thus, we fitted an ellipse [12] to the capacitive images to estimate the pitch angle using Boring et al.'s approach [3]. The initial visual inspections looked promising for an ellipse to determine the pitch angle, see Figure 3.

Boring et al.'s approach [3] assumes a linear relationship between the ellipse major and the pitch of the finger. Therefore, we fitted a line to the ellipse major to understand if this is a good approximation for the thumb. The linear fit for the ellipse major has a R^2 of .86, see Figure 4. This suggests an overall low-fitting quality. Using this linear fit, we can furthermore determine the error on the test set. The fit results in a RMSE of 17.2° with a mean absolute error of 13.8° ($SD = 10.2°$), see Estimation Error in Figure 4.

4.3 Model Training

The goal of our model is to predict the thumb's pitch based on the capacitive image; hence, the input is the capacitive image. During model training, we used the ground truth pitch angle that we recorded using the motion tracking system in our data collection study. As data representation approaches such as CNNs have been shown to be more effective for capacitive data than feature extraction approaches, e.g., [24, 27, 34], we opted to skip this step and started exploring deep CNN models.

We used the trial-and-error method [7] combined with a grid search for hyper-parameters tuning using 8 : 5 : 3 participants, a common split of about 50%:30%:20%, for the training, testing, and validation set, respectively. The validation set remains untouched during the training phase. After testing a wide range of different architectures, we determined that a CNN with three convolution layers and two dense layers yields the best results. The final model structure and the network parameters are depicted in Figure 5. The dropout layers are set to 0.5 and the CNN kernel size to 3×3. For all layers, we used the ReLU activation function. Additionally, we found that L2 regularizers after each layer with a value of 0.4 perform the best.

As the loss function, we decided to use a mean-square error (MSE) function to reduce potential overfitting toward the outlier. While testing different optimizers, we found that training with an Adam optimizer yielded the best results. Additionally, we settled on the following parameters for the optimizer after testing their impact on the training results: we settled on a learning rate starting from .001 with a reduction by 5% after 10,000 epochs without improvement and a minimal learning rate of .00001. We used a batch size of 2,000 and trained the model for 300,000 epochs, which took approximately nine days on an NVIDIA Tesla V100 to train.

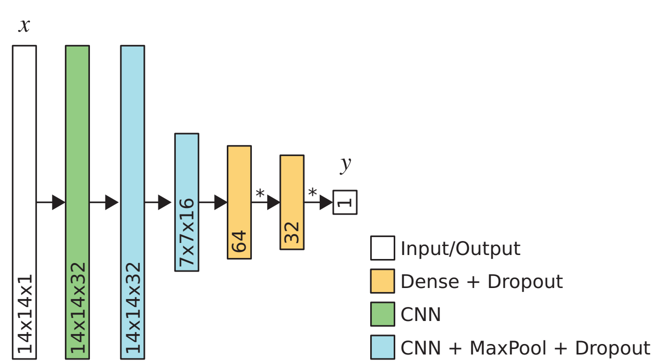


Figure 5: An illustration of the architecture of our CNN regression model to estimate the pitch value.

Afterward, we performed a warm-start to push our initial results, a common technique to optimize deep-learning ML models. For the warm-start training phase, we used an early stopping approach. The training stopped after an additional 98,945 epochs. We expected such a long training time due to the low learning rate; however, we found this leads to more stable results.

4.4 Model Results

Our regression model estimates the thumb's pitch angle between 0° (flat) and 90° (steep) based on the raw capacitive image. Our final model estimates the thumb's pitch angle with a *RMSE* of 8.2° (*MAE* = 6.4°, *SD* = 5.1°) on the training set, see Table 1. The model is accounting for similar results for the test set when keeping in mind that the test set is not artificially augmented: *RMSE* = 9.8° (*MAE* = 7.5°, *SD* = 6.3°). Finally, our model achieved an accuracy on the untouched validation set with *RMSE* = 11.7° (*MAE* = 9.5°, *SD* = 6.9°).

4.5 Time Performance

We carried out a performance test on a Nexus 5 phone to determine the prediction time. We ran 4,000 predictions and measured the time for the preprocessing and the model prediction itself. The preprocessing took an average of 10.45 ms (*SD* = 2.51), and the model itself another 14.09 ms (*SD* = 3.51). Because we receive the capacitive images every 50 ms (20 FPS), we can estimate the thumb's pitch for each retrieved capacitive image, allowing us to run *ThumbPitch* at maximum performance.

5 USE CASES

We envision that *ThumbPitch* will enrich touch-based interaction for a wide range of scenarios. We argue that *ThumbPitch* can overcome the limitations of previously proposed interaction techniques and, for the first time, support not only two-handed but also one-handed interactions. Moreover, *ThumbPitch* supports a large input range compared to 2-level input, such as ForceTouch by Apple and KnuckleTouch [34]. This allows the user to perform *ThumbPitch* in both one-handed and two-handed scenarios, giving the user the option to use both hands if possible but also enabling interaction in mobile and encumbered situations, e.g., while carrying a shopping bag. In the following, we present several use cases in which we envision *ThumbPitch* to outperform previously proposed methods to highlight the possibilities in user interface design.

Table 1: The baseline and model fitting results. All values are in degrees.

	RMSE	MAE	SD
Baseline by Boring et al. [3]	15.5	12.6	8.9
Our CNN Model – Train Set	8.2	6.4	5.1
Our CNN Model – Test Set	9.8	7.5	6.3
Our CNN Model – Validation Set	11.7	9.5	6.9

5.1 Sliders

Sliders often span across the whole screen and can be cumbersome for users as they need to stretch their thumb across the whole screen. Here, we envision *ThumbPitch* to be an additional input method to change the slider value. In detail, we envision the thumb's angle to be mapped to the slider position. Thus, no change in position needs to be performed by the user; a simple change in angle will modify the position and, thus, the input.

5.2 Zoom

Pinch-to-zoom has become a ubiquitous gesture to zoom into maps, images, and various other content. However, the gesture to zoom requires two fingers on the screen, which is nearly impossible in a one-handed interaction. Here, we envision *ThumbPitch* can substitute the traditional pinch-to-zoom gesture, enabling easy zoom interaction even in encumbered and mobile scenarios.

5.3 Drawing

Today, the user either needs external hardware (e.g., Apple Pencil or Microsoft Surface Dial) or needs to change the settings through the user interface provided by the drawing app for rich input such as different stroke colors or stroke width. With *ThumbPitch*, we enable an enriched input that allows the user to change various parameters on the fly. For instance, we envision *ThumbPitch* to change the stroke color or stroke width as the thumb angle can be mapped to the width or color wheel (see Figure 6).

5.4 Context Menu Selection

As phone sizes are increasing, reachability issues are getting more prominent. We envision an implementation where the thumb's up and down movement can be used to scroll through the content on the screen. Thus, *ThumbPitch* allows scrolling through menu items and selecting an option on release without encountering the reachability issues associated with larger phones. Moreover, it allows convenient scrolling on devices with small screens, such as smartwatches.

5.5 Up/Down Pitch Gesture

ThumbPitch can also offer extra dimensions for gestures. As the thumb range is limited [25], performing gestures with the thumb can be difficult, especially in one-handed scenarios. Thus, *ThumbPitch* can enrich the gestures' limited vocabulary.

6 EVALUATION STUDY

In this next section, we evaluate *ThumbPitch* as an additional input dimension to enrich touch-based interactions. Here, we compared *ThumbPitch* against the standard *Touch* input. Therefore, we implemented three use-cases, which allowed us to acquire in-situ feedback about using *ThumbPitch* in *Drawing*, *MapZoom*, and *Slider* tasks.

6.1 Apparatus

For the first part of the study, we used the same setup as in the first study: an LG Nexus 5 for capacitive recording and an OptiTrack system for ground truth touch collection.

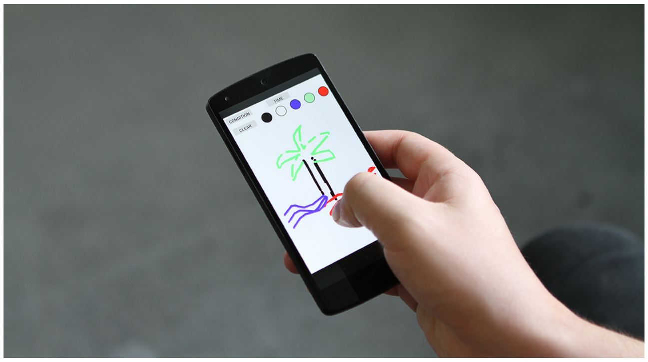


Figure 6: A participant drawing in the validation study using *ThumbPitch*.

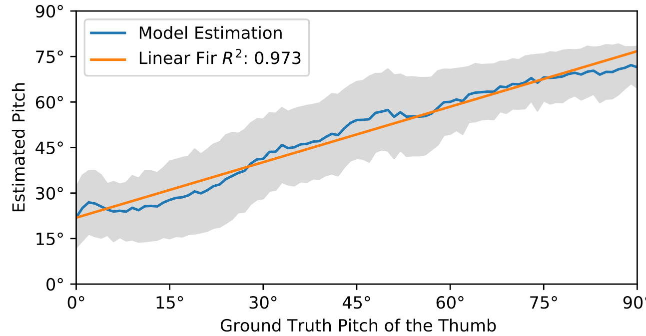


Figure 7: The estimation error for our model in the evaluation study with 12 new participants.

For the second part, we not only gathered the capacitive images but also fed them into our new model to predict the thumb's pitch. This allows us to implement three use cases: *Drawing*, *MapZoom*, and *Slider*. We implemented all tasks with *ThumbPitch* (*Pitch* condition) as well as with standard touch interaction (*Touch* condition).

6.2 Tasks

In the *Drawing* task, we used the thumb's pitch as an additional input to change the stroke width. Here, high pitch values (steep finger) result in a wide stroke, and low values (flat finger) in a thin line. In the *Touch* condition, we added extra buttons to change the stroke width (see Figure 6).

In the *MapZoom* task, we asked participants to zoom into three different cities of their own choice. We implemented the task in a way that the thumb's pitch zooms the map, and the thumb's movement in the x and y directions changes the map position. Higher pitch values zoom into the map, while lower pitch values zoom out.

In the *Slider* use case, we asked participants to manipulate the slider value using the thumb pitch. In detail, the app showed a number between 0 and 20, and participants were asked to set the slider to the same value. The slider was positioned horizontally, and the lower values were located on the left end of the slider. Lower pitch values moved the slider further left and higher pitch values further right.

6.3 Procedure

We welcomed the participants and explained the study to them in detail. After answering any questions, we asked them to sign an informed consent form. We then attached the markers for the first part. Participants were seated during the whole study.

For the first part, we followed the procedure of the first study. However, as we wanted to validate the model, we gave no other instruction other than to vary the pitch as they liked, but if possible, to explore the full pitch range. After five minutes of collecting validation data, we moved on to the second part of the study.

In the second part, the participants started either in the *Touch* or *Pitch* condition; the order of the conditions was Latin Square balanced. Within each condition, participants were asked to perform the three tasks (*Drawing*, *MapZoom*, and *Slider*) in random order. We asked participants to draw a scene of their vacation experiences, zoom in on three different cities, and put in five values using the slider. After each condition, participants were asked to fill in a raw NASA-TLX [14] and a system usability scale (SUS) [4]. We collected feedback on perceived ease, speed, success, accuracy, and comfort on a 7-point Likert scale [34].

Finally, we interviewed participants to understand how they liked the idea of *ThumbPitch* overall. Afterward, we thanked the participants again and paid 10 EUR per hour. The study took an average of 45 minutes.

6.4 Participants

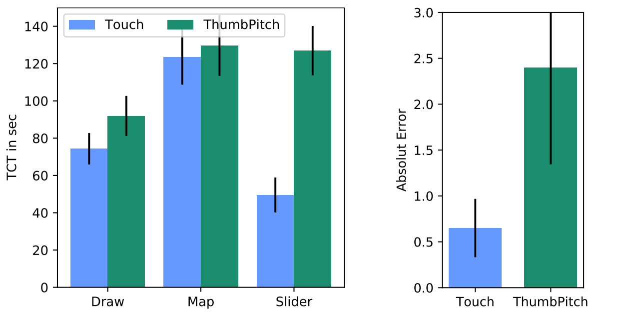
We recruited 12 new participants from an internal university volunteer pool who did not participate in the first study. The participants (six female and six male) were between 21 and 29 years old (M = 24.2, SD = 2.5). No participant had any movement impairments. As in our data collection study, only right-handed participants took part in this study.

6.5 Results

In the following, we present the quantitative results and subjective feedback of the *ThumbPitch* evaluation study. We conducted the study with participants who did not participate in the data collection study to avoid overfitting.

6.5.1 Model Accuracy. We gathered in total of 34,517 new capacitive images for validation from the 12 new participants in the evaluation study. The model on the validation dataset from the second study has a root-mean-square error (RMSE) of 15.38° (MAE = 11.90°, SD = 9.75°). For comparison, our results are in line with the angular error results for finger orientation input (e.g., [27, 38]). Figure 7 shows the estimation for the different inputs. The model's estimation can be described as a monotonic growth (linear fit R^2 = .973), see Figure 7. This improves over our estimation using the contact size, see Figure 4. However, the model under- or over-estimates the pitch for the close to flat (0°) and the close to steep angles (90°), respectively.

6.5.2 Task Completion Time (TCT). Due to the nature of the tasks being vastly different, we conducted three independent tests for the TCT, see Figure 8a. We log-transformed all times before performing



(a) TCT (b) Error

Figure 8: a) shows the TCT for the three different tasks. b) shows the absolute errors on the Slider input.

a t-test as three Shapiro-Wilk tests revealed that the times are normally distributed (*Slider*: $W = .957$, $p = .381$; *Draw*: $W = .962$, $p = .499$; and *Map*: $W = .946$, $p = .218$). The t-test revealed only a significant difference for *Slider* ($t(11) = -6.207$, $p < 0.001$). On the other hand, both *Draw* and *Map* tasks showed no significant difference ($t(11) = -1.731$, $p = .112$; $t(11) = -0.267$, $p = .794$, respectively).

6.5.3 Slider Error. The results show that the slider error was larger when using *ThumbPitch*. The Shapiro-Wilk tests revealed that the data were not normally distributed; thus, we performed a Wilcoxon signed-rank test ($W = .853$, $p < .003$). The test revealed that touch is significantly less error-prone in comparison to *ThumbPitch* ($V = 0$, $p < 0.004$). Thus, we conclude that *Touch* ($M = .6$, $SD = 1.1$) is more precise than *ThumbPitch* ($M = 2.4$, $SD = 3.7$), see Figure 8b.

6.5.4 Subjective Feedback in Realistic Scenarios. In the three scenarios of the evaluation study, we asked participants to complete an SUS questionnaire [4], raw NASA-TLX questionnaire [14], and the subjective perception by Le et al. [22]. The qualitative results

Table 2: The quantitative results for the three tasks split by condition: *Touch* and *ThumbPitch*. Raw NASA-TLX on a 21-point scale (0-20), the SUS on its' standard scale from 0 to 100, and the subjective perceptions (7-point Likert scale) as described by Le et al. [22]. The results of the normality test (Shapiro-Wilk-Tests) as justification for t-tests, as well as t and p-values, are present.

	Touch		*ThumbPitch*		*norm.*		*t-test*	
	M	SD	M	SD	W	p	t(11)	p
TLX [14]	4.6	3.8	9.2	4.6	.930	.098	−4.490	<.001
SUS [4]	77.4	24.4	54.1	25.7	.976	.802	4.185	<.002
Easiness	5.5	1.8	4.0	1.7	.946	.221	2.955	<.014
Speed	5.5	1.8	3.6	1.6	.927	.083	4.667	<.001
Success	5.3	1.9	3.4	1.7	.952	.307	4.703	<.001
Accuracy	4.8	2.1	3.1	1.7	.928	.088	4.330	<.001
Comfort	5.4	1.6	3.8	1.7	.940	.169	4.020	<.002

overall showed that normal touch input was easier to perform than the *ThumbPitch*, see Table 2.

Eight participants rated in favor of *ThumbPitch* while four were conservative about having *ThumbPitch* in their next device (Pro: P1, P4 - P7, P9 - P11 vs. Con: P2, P3, P8, P12). The favorite "killer" feature was the map zoom capabilities (P1, P3, P5 - P7, P9 - P11). Participants reported that *ThumbPitch* is intuitive (P11), attractive (P1, P4, P7), and easy (P1, P4, P6 - P8). P5 summarizes that "you can use your phone with only one hand." Four participants commented on the extended possibility of controlling their devices, allowing them to use their devices better (P1, P3, P5, P11). Five participants (P1, P5, P9 - P11) said that they would not use it in all situations as the input only works with the thumb. While this is true, this is the purpose of *ThumbPitch*. Here, the participants picked up on a second intended feature of *ThumbPitch*, as they commented on the improved ergonomics. In detail, they said that *ThumbPitch* allows them to move the thumb less across the screen (P1, P2, P8, P10). Six participants said that they were unfamiliar with *ThumbPitch* and, therefore, it was difficult to perform at first but they think they could adjust to the input (P2 - P4, P6, P7, P10). For P8, for instance, this was a reason not to like ThumbPitch: "I'm not used to using my thumb, so I didn't like it very much." Additionally, we received comments regarding poor accuracy (P5 - P9). Three participants (P2, P3, P11) did not like the implementation of our use case.

Finally, our participants envisioned scrolling through word selection and other keyboard operations (P1, P4, P8). More specific use cases were for games, e.g., Tetris (P6, P10), and general scrolling (P5). Finally, participants envisioned *ThumbPitch* for gestures, e.g., locking the screen and volume control (P10 - P12).

7 DISCUSSION

We conceptualized, implemented, and evaluated *ThumbPitch* to enrich one-handed touchscreen interactions. Hereby, we extend over the existing FatThumb [3] approach, which relays on the ellipse major. With *ThumbPitch*, we can determine the pitch of the finger in a more precise manner by using raw capacitive images. Moreover, in contrast to the FatThumb approach [3], which needs a one-time calibration step, our accuracy results show that our approach generalizes across participants. Moreover, we can estimate the full 90° input, allowing for continuous input instead of a stepped approach, which is mostly limited to two or three different input categories. Here, we have a smooth transition between neighboring inputs, highlighted by an R^2 value of .97 (see Figure 7) in contrast to the baseline approach, $R^2 = .69$ (see Figure 4). Thus, our approach results in a smoother input and no sudden jumps in pitch angle estimation.

Overall, we received positive responses from the evaluation study, as eight participants would want to have this feature on their next phone. However, as expected, the quantitative feedback did not turn out in favor of *ThumbPitch*; this is in line with related work introducing new interaction techniques (e.g., [22, 34]). This was further addressed by participants of the second study, where they mentioned that they were unfamiliar with using *ThumbPitch* and would require more time to become accustomed to the new interaction possibility. This impacts the results compared to touch interaction, which all users use every day on their devices. Moreover, while the SUS seems to be low initially, we stress that it is essential to compare it to the traditional baseline, which is low. Thus, we argue that this indicates that the participants rated the task itself (draw, map, and slider), which impacted the SUS results even more. While the draw and map tasks resulted in similar results for TCT, the *ThumbPitch* slider did not follow this trend. After analyzing the differences, the main problem was the lift-off after the participants selected the right position. Our current ML model does not adjust for the touch-up phase. However, we argue that a simple time offset could already reduce this issue. Alternatively, as the capacitive matrix is already exposed to the model, it should be extended to an LSTM model, e.g., [34]), which can then also detect touch-up events for *ThumbPitch* input.

Considering only the accuracy, our evaluation study with 12 new participants revealed that the *MAE* was only 0.20° larger than the validation dataset of the first study. Given that the validation dataset contains only three participants and the new dataset contains 12 different participants, we argue that the error is consistent, and the ML model is not subject to overfitting. In terms of accuracy, we achieved a remaining error of around 10°; this error is similar to the error for finger orientation input (e.g., [27, 38]). While the absolute error is an impotent indicator, if a system can work ultimately, then only usability testing can determine if users can interact with a system. Therefore, we conducted a validation study as the second part and included three specific tasks to get the users' feedback. We can confirm that participants could perform different tasks with the given accuracy.

With *ThumbPitch*, the fear of dropping the phone arises, especially as phones become more expensive. However, Bergstrom-Lehtovirta and Oulasvirta [1] already showed a large interaction range of the thumb on the screen, and Le et al. [24] extend this finding for one-handed scenarios while walking. Thus, we argue that the risk will be minimal. However, while studying ergonomic constraints of *ThumbPitch* is outside of this paper's scope, we intend to formally study its ergonomic effects and the risk of the device dropping in the future.

8 OPEN SOURCE

To permit others to use our implementation, as well as facilitate replication and others wishing to explore and extend our approach, we have open-sourced our datasets, model, and training script at https://github.com/sven-mayer/thumbpitch.

9 CONCLUSION

This paper presented *ThumbPitch*, a continuous input dimension for touch-based devices that targets especially one-handed interaction scenarios. We achieve this by utilizing additional information from capacitive images, which today's touch controllers ignore. This allows deploying our model on today's touch devices through a simple software update. Further, we provide a ready-to-deploy machine learning model to estimate the thumb pitch angle. This allows other researchers to build on this work directly.

In the future, we plan to run an in-the-wild study in which we hand out *ThumbPitch*-enabled phones to new participants. As we cannot modify in-app functions, we plan to deploy a gesture layer on top of the operating system to enhance interaction. However, in the long-term, we hope that manufacturers adopt the feature as we found that *ThumbPitch* offers a true alternative to pinch-to-zoom and offers a high potential to enhance touch interaction in the future.

ACKNOWLEDGMENTS

This work has been partly funded by the German Federal Ministry of Education and Research and the Bavarian State Ministry for Science and the Arts through https://www.mcml.ai/. The authors of this work take full responsibility for its content.

REFERENCES

[1] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. 2014. Modeling the Functional Area of the Thumb on Mobile Touchscreen Surfaces. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 1991–2000. https://doi.org/10.1145/2556288.2557354

[2] Tobias Boceck, Sascha Sprott, Huy Viet Le, and Sven Mayer. 2019. Force Touch Detection on Capacitive Sensors using Deep Neural Networks. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'19)*. ACM, New York, NY, USA, 6. https://doi.org/10.1145/3338286.3344389

[3] Sebastian Boring, David Ledo, Xiang Anthony Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (San Francisco, California, USA) *(MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/2371574.2371582

[4] John Brooke. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[5] Drini Cami, Fabrice Matulic, Richard G. Calland, Brian Vogel, and Daniel Vogel. 2018. Unimanual Pen+Touch Input Using Variations of Precision Grip Postures. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. ACM, New York, NY, USA, 825–837. https://doi.org/10.1145/3242587.3242652

[6] Ashley Colley and Jonna Häkkilä. 2014. Exploring Finger Specific Touch Screen Interaction for Mobile Phone User Interfaces. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design* (Sydney, New South Wales, Australia) *(OzCHI '14)*. ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2686612.2686699

[7] Paulin Coulibaly, François Anctil, and Bernard Bobée. 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology* 230, 3 (2000), 244–257. https://doi.org/10.1016/S0022-1694(00)00214-6

[8] Sanjay Ghosh, Joy Bose, Rajkumar Darbar, and Punyashlok Dash. 2017. Touch Shadow Interaction and Continuous Directional User Interface for Smartphone. In *Human-Computer Interaction - INTERACT 2017*. Springer International Publishing, Cham, 66–74.

[9] Hyunjae Gil, Hongmin Kim, and Ian Oakley. 2018. Fingers and Angles: Exploring the Comfort of Touch Input on Smartwatches. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 164 (Dec. 2018), 21 pages. https://doi.org/10.1145/3287042

[10] Hyunjae Gil, DoYoung Lee, Seunggyu Im, and Ian Oakley. 2017. TriTap: Identifying Finger Touches on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 3879–3890. https://doi.org/10.1145/3025453.3025561

[11] Anhong Guo, Robert Xiao, and Chris Harrison. 2015. CapAuth: Identifying and Differentiating User Handprints on Commodity Capacitive Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. ACM, New York, NY, USA, 59–62. https://doi.org/10.1145/2817721.2817722

[12] Radim Halır and Jan Flusser. 1998. Numerically stable direct least squares fitting of ellipses. In *Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization (WSCG '98, Vol. 98)*. WSCG, Plzen - Bory, Czech Republic, 125–132.

[13] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. ACM, New York, NY, USA, 627–636. https://doi.org/10.1145/2047196.2047279

[14] Sandra G. Hart. 2006. NASA-Task Load Index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomic Society annual meeting* 50, 9 (2006),

904–908. https://doi.org/10.1177/154193120605000909

[15] Christian Holz and Patrick Baudisch. 2010. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. ACM, New York, NY, USA, 581–590. https://doi.org/10.1145/1753326.1753413

[16] Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. ACM, New York, NY, USA, 2501–2510. https://doi.org/10.1145/1978942.1979308

[17] Christian Holz, Senaka Buthpitiya, and Marius Knaust. 2015. Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 3011–3014. https://doi.org/10.1145/2702123.2702518

[18] Osman Semih Kayhan and Jan C. van Gemert. 2020. On Translation Invariance in CNNs: Convolutional Layers Can Exploit Absolute Spatial Location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Manhattan, New York, US, 14262–14273. https://doi.org/10.1109/CVPR42600.2020.01428

[19] Jiwan Kim and Ian Oakley. 2022. SonarID: Using Sonar to Identify Fingers on a Smartwatch. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 287, 10 pages. https://doi.org/10.1145/3491102.3501935

[20] Sven Kratz, Patrick Chiu, and Maribeth Back. 2013. PointPose: Finger Pose Estimation for Touch Input on Mobile Devices Using a Depth Sensor. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (St. Andrews, Scotland, United Kingdom) *(ITS '13)*. ACM, New York, NY, USA, 223–230. https://doi.org/10.1145/2512349.2512824

[21] Abinaya Kumar, Aishwarya Radjesh, Sven Mayer, and Huy Viet Le. 2019. Improving the Input Accuracy of Touchscreens Using Deep Learning. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. ACM, New York, NY, USA, Article LBW1514, 6 pages. https://doi.org/10.1145/3290607.3312928

[22] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm As an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 360, 13 pages. https://doi.org/10.1145/3173574.3173934

[23] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2017. A Smartphone Prototype for Touch Interaction on the Whole Device Surface. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. ACM, New York, NY, USA, Article 100, 8 pages. https://doi.org/10.1145/3098279.3122143

[24] Huy Viet Le, Sven Mayer, and Niels Henze. 2019. Investigating the Feasibility of Finger Identification on Capacitive Touchscreens Using Deep Learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. ACM, New York, NY, USA, 637–649. https://doi.org/10.1145/3301275.3302295

[25] Huy Viet Le, Sven Mayer, Benedict Steuerlein, and Niels Henze. 2019. Investigating Unintended Inputs for One-Handed Touch Interaction Beyond the Touchscreen. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 34, 14 pages. https://doi.org/10.1145/3338286.3340145

[26] Sven Mayer, Perihan Gad, Katrin Wolf, Paweł W. Woźniak, and Niels Henze. 2017. Understanding the Ergonomic Constraints in Designing for Touch Surfaces. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. ACM, New York, NY, USA, Article 33, 9 pages. https://doi.org/10.1145/3098279.3098537

[27] Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. ACM, New York, NY, USA, 220–229. https://doi.org/10.1145/3132272.3134130

[28] Sven Mayer, Huy Viet Le, and Niels Henze. 2018. Designing Finger Orientation Input for Mobile Touchscreens. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Barcelona, Spain) *(MobileHCI '18)*. ACM, New York, NY, USA, Article 29, 9 pages. https://doi.org/10.1145/3229434.3229444

[29] Sven Mayer, Xiangyu Xu, and Chris Harrison. 2021. Super-Resolution Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 12, 10 pages. https://doi.org/10.1145/3411764.3445703

[30] Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD. Rasel Islam. 2016. The Flat Finger: Exploring Area Touches on Smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (Santa Clara, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 4238–4249. https://doi.org/10.1145/2858036.2858179

[31] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2011. AnglePose: Robust, Precise Capacitive Touch Tracking via 3d Orientation Estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2575–2584. https://doi.org/10.1145/1978942.1979318

[32] Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 927–936. https://doi.org/10.1145/1518701.1518843

[33] Martin Schmitz, Florian Müller, Max Mühlhäuser, Jan Riemann, and Huy Viet Viet Le. 2021. Itsy-Bits: Fabrication and Recognition of 3D-Printed Tangibles with Small Footprints on Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 419, 12 pages. https://doi.org/10.1145/3411764.3445502

[34] Robin Schweigert, Jan Leusmann, Simon Hagenmayer, Maximilian Weiß, Huy Viet Le, Sven Mayer, and Andreas Bulling. 2019. KnuckleTouch: Enabling Knuckle Gestures on Capacitive Touchscreens Using Deep Learning. In *Proceedings of Mensch Und Computer 2019* (Hamburg, Germany) *(MuC'19)*. Association for Computing Machinery, New York, NY, USA, 387–397. https://doi.org/10.1145/3340764.3340767

[35] Matthieu B. Trudeau, Justin G. Young, Devin L. Jindrich, and Jack T. Dennerlein. 2012. Thumb motor performance varies with thumb and wrist posture during single-handed mobile phone use. *Journal of Biomechanics* 45, 14 (2012), 2349–2354. https://doi.org/10.1016/j.jbiomech.2012.07.012

[36] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. 2009. Detecting and Leveraging Finger Orientation for Interaction with Direct-touch Surfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (Victoria, BC, Canada) *(UIST '09)*. ACM, New York, NY, USA, 23–32. https://doi.org/10.1145/1622176.1622182

[37] Katrin Wolf, Markus Schneider, John Mercouris, and Christopher-Eyk Hrabia. 2015. Biomechanics of front and back-of-tablet pointing with grasping hands. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 7, 2 (2015), 43–64. https://doi.org/10.4018/ijmhci.2015040103

[38] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. ACM, New York, NY, USA, 47–50. https://doi.org/10.1145/2817721.2817737

[39] Vadim Zaliva. 2012. 3D finger posture detection and gesture recognition on touch surfaces. In *12th International Conference on Control Automation Robotics Vision (ICARCV 2012, December)*. IEEE, IEEE, Manhattan, New York, US, 359–364. https://doi.org/10.1109/ICARCV.2012.6485185

[40] Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, and Ken Hinckley. 2019. Sensing Posture-Aware Pen+Touch Interaction on Tablets. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 55, 14 pages. https://doi.org/10.1145/3290605.3300285