

# Deep Touch: Sensing Press Gestures from Touch Image Sequences

Philip Quinn, Wenxin Feng, and Shumin Zhai

**Abstract** Capacitive touch sensors capture a sequence of images of a finger’s interaction with a surface that contain information about its contact shape, posture, and biomechanical structure. These images are typically reduced to two-dimensional points, with the remaining data discarded – restricting the expressivity that can be captured to discriminate a user’s touch intent. We develop a *deep touch* hypothesis that (1) the human finger performs richer expressions on a touch surface than simple pointing; (2) such expressions are manifest in touch sensor image sequences due to finger–surface biomechanics; and (3) modern neural networks are capable of discriminating touch gestures using these sequences. In particular, a *press* gesture based on an increase in a finger’s force can be sensed without additional hardware, and reliably discriminated from other common expressions. This work demonstrates that combining capacitive touch sensing with modern neural network algorithms is a practical direction to improve the usability and expressivity of touch-based user interfaces.

## 1 Introduction

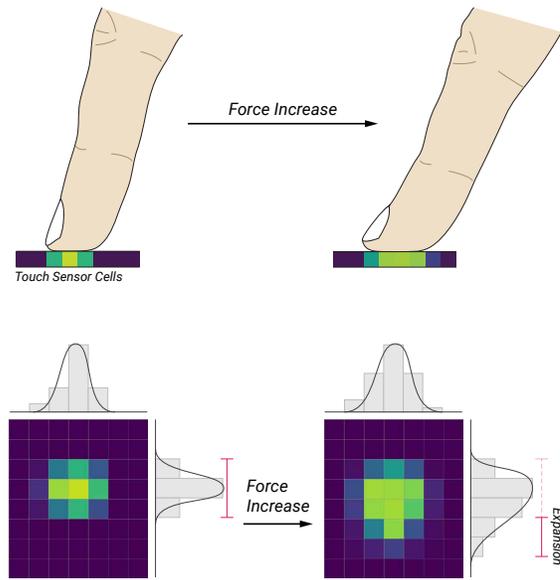
Touch interaction is predominantly driven by two-dimensional pointing – where a user’s contact on a surface is reduced to a single point (its *centroid*), with criteria

---

Philip Quinn (e-mail: philip@quinn.gen.nz) · Wenxin Feng (e-mail: wenxinfeng@google.com) · Shumin Zhai (e-mail: zhai@acm.org)  
Google, Mountain View, CA

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021  
Y. Li and O. Hilliges (eds.), *Artificial Intelligence for Human Computer Interaction: A Modern Approach, Human–Computer Interaction Series*

**Fig. 1** An illustration of a *press* gesture: a user's finger contacts a touch sensor and deforms as the force behind it increases (top); this deformation is observed as a unilateral expansion on the sensor image (bottom).



placed on its spatiotemporal properties to define various gestures (Buxton et al, 1985; Buxton, 1995). The most common of these gestures in contemporary touch interaction are *tapping*, *long pressing (touch and hold)*, and *scrolling* (panning, dragging, flicking, and surface-stroke gestures, etc.), which are modelled using a set of three heuristics: (1) if the distance from the initial contact location exceeds a hysteresis threshold, the gesture is a *scroll*; (2) if the duration since the initial contact exceeds a time threshold, the gesture is a *long press*; (3) otherwise, the gesture is a *tap* when the finger contact is lifted. Although this model has nurtured a broad and successful design space for interaction, it belies the rich signal that touch sensors produce. In particular, capacitive touch sensors capture an ‘image’ of the finger’s contact shape that can reveal the evolution of a finger’s posture during a contact (Figure 1).

As *long press* relies on a latency threshold, it is the least direct, discoverable, usable, or expressive of the three gestures described. Force sensing has been explored – both academically and commercially – as a method for rectifying these problems by creating a *force press* gesture that is directly connected to an active parameter of the user’s input: their finger’s force. However, force sensing requires additional hardware that suffers from practical challenges in its cost and integration.

Observations and analyses of the human finger’s biomechanics and the underlying touch sensor technology (capacitive sensing) suggest a complementary approach to the latency-based *long press*. In many cases, a press gesture is manifest in the subtle signals that are captured by the image sequence from a capacitive touch sensor: as a user increases their finger’s force on a screen, its contact mass increases unilaterally

as the strain on the most distal finger joint increases and the finger rolls downward (Figures 1 and 6).

These raw images are difficult to analyse heuristically due to the high dimensionality of the data, temporal dependencies in the gesture, sensor noise under different environmental conditions, and the range of finger sizes and postures that may be used. However, modern neural learning techniques present an opportunity to analyse touch sensor images with a data-driven approach to classification that is robust to these variances.

We call this approach *deep touch*: a neural model for sensing touch gestures based on the biomechanical interaction of a user’s finger with a touch sensor. To differentiate between gestures, we use a neural network to combine complex spatial (convolutional) features from individual images with temporal (recurrent) features across a sequence of touch sensor images. We identify a set of biomechanical patterns to shape the learned features and minimise the number of parameters so that the resultant model can be used in real-time without impairing system latency or responsiveness. Although this does not allow force sensing per se, it can recognise a user’s intention to *press* as a discrete gesture.

In this chapter, we first present an overview of touch sensing hardware, finger–surface interaction, and the touch input design space. The overview highlights a weakness of the current touch interaction system: the lack of a direct *pressing* gesture. We then describe the deep touch model: the biomechanical patterns, neural model design, data collection methodology, and training procedure. Finally, we describe how this model was integrated into the Android gesture classification algorithm as part of Google Pixel 4 and 5 devices without incurring additional input latency.

## 2 Touch Sensing and Finger Interaction

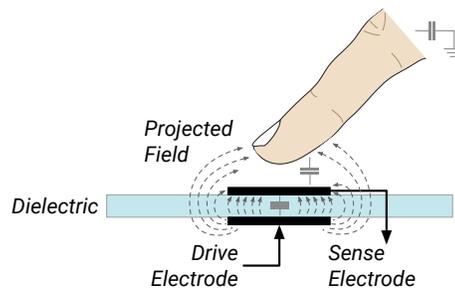
### 2.1 Touch-Sensing Hardware

Of the techniques for detecting the presence of a human finger near an object (reviewed by Zimmerman et al, 1995; Walker, 2012; Grosse-Puppendahl et al, 2017), the most common for small–medium size mobile devices is Projected Capacitive Touch (PCT or ‘p-cap’). PCT is based on the principle of *capacitive coupling* (Barrett and Omote, 2010; Walker, 2012): when two conductive objects (*electrodes*) are brought close together, they can hold a charge between them – their *capacitance* – which becomes disrupted when another conductive object encroaches. The capacitance  $C$  of two such electrodes, separated by a dielectric material (usually glass or plastic), is given by

$$C = \epsilon_k \epsilon_0 \frac{A}{d}, \quad (1)$$

where  $\epsilon_k$  is the dielectric constant of the separating material,  $\epsilon_0$  is the constant permittivity of a vacuum,  $A$  is the area of overlap between the electrodes, and  $d$  is

**Fig. 2** A projected capacitive touch (PCT) sensor: two electrodes separated by a dielectric. A capacitive coupling is created with a field projected from a drive electrode and measured on a sense electrode. This field is disrupted when another conductive object (e.g. a finger) comes close.



the distance between them. When one electrode is driven (the transmitting electrode), it projects an electric field that allows  $C$  to be measured on the other (the sensing electrode). When another conductive object approaches this field, as in Figure 2, it ‘steals’ some of the charge from the field by shunting it to ground through its own inherent capacitance path – reducing the steady-state  $C$  by some amount ( $\Delta C$ ).<sup>1</sup> For a human finger this is typically on the order of 100 pF.

In a touch sensor on a display these electrodes are typically made from a transparent conductor (such as indium tin oxide – ITO), and arranged as a layer of rows and a layer of columns, with the dielectric sandwiched between them (Westerman, 1999; Lee et al, 2014). The locations where the electrodes overlap (the *cells*) are where  $\Delta C$  is measured.

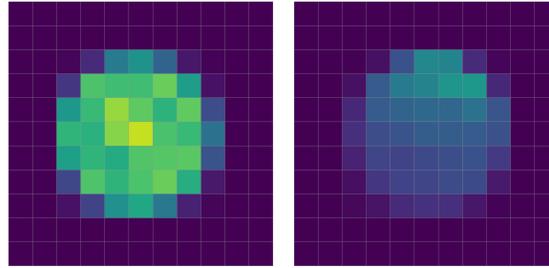
This matrix of electrodes has a much lower resolution than the pixel matrix of the underlying display – for example, the Google Pixel 4 has pixel density of roughly 20 px/mm, but a touch cell density of roughly 0.25 cells/mm. This discrepancy is resolved by a *touch controller* that performs several processing steps (reviewed by Wang and Blankenship, 2011; O’Connor, 2010)—

1. *Baselining* to remove the steady-state capacitance  $C$  when there are no touch contacts.
2. *Filtering* to remove analogue noise (e.g. from the components driving the display) and parasitic capacitance.
3. *Centroiding* to segment and interpolate the cell responses to a set of contacts and their precise locations.

Figure 3 (left) shows an example of a touch sensor’s response to a 25 mm metal coin after baselining and filtering. The touch sensor produces a signal that is concentrated at the centre of the coin, and with fringe fields that extend slightly beyond its bounds. The signal’s high sensitivity to minute changes in an object’s position allows a centroid to be resolved with sub-pixel precision.

Touch sensors are protected by a covering glass and are tuned to maximise their sensitivity to objects touching this glass – but they do not require an object to be

<sup>1</sup> Sensing  $\Delta C$  is known as sensing the *mutual capacitance* between electrodes. In a related (but more limited) technique, the *self capacitance*  $C$  of each electrode is measured individually (Barrett and Omote, 2010; Walker, 2012).



**Fig. 3** A touch sensor’s response to a 25 mm metal coin (left) and with a 0.5 mm plastic shim under its base (right). Each cell is 4.5 mm<sup>2</sup>; brighter cells indicate higher  $\Delta C$  values.

in direct contact to produce a signal. Figure 3 (right) shows the sensor’s response with a plastic shim lifting the coin at a small angle. The signal produced is a smooth gradient as the distance between the touch sensor and the coin increases. Some sensor designs can amplify this effect for sensing objects that are up to 30 cm from the sensor (e.g. Hu et al, 2014).

PCT does not inherently detect the force applied to a sensor as changes in force do not normally alter the capacitive properties of an object. That is, orthogonal forces applied to the finger in Figure 2 will not change the  $\Delta C$ .<sup>2</sup> Rather, the force on a touch surface is typically measured using a layer of Force Sensing Resistors (FSRs) or strain gauges that change their electrical resistance with forces upon them that deform the surface (Yaniger, 1991; Rosenberg and Perlin, 2009).

## 2.2 Finger–Surface Biomechanics

A finger’s contact with a touch screen is not a rigid-body interaction. The fingertip is a soft, compliant object with complex dynamic properties. These properties produce consequential dynamic changes to the signal observed by a touch sensor when a fingertip is pressed against it.

The fingertip consists of a distal phalanx bone, wrapped by a *fingertip pulp* that is mostly composed of subcutaneous fat, and a skin membrane (Figure 4). The fingertip pulp has the properties of a viscoelastic material when it is pressed against a surface: responses are repeatable, but have hysteresis, rate-dependence, and non-linear effects (Serina et al, 1997; Srinivasan et al, 1992; Srinivasan and LaMotte, 1995; Pawluk and Howe, 1999; Miyata et al, 2007).

Serina and colleagues (1997; 1998) measured and modelled the vertical compression and contact area responses of a fingertip pressing against a flat surface at different angles and forces. They found the fingertip pulp was very responsive to changes at force levels under 1 N, and quickly saturated at higher levels (e.g. 69%

<sup>2</sup> If the electrodes are allowed the ‘float’ with respect to each other, then changes in the distance between them from external forces can be detected by Equation 1.

of the contact area at 5.2 N of force was achieved by 1 N). The effects were robust across angles, and were invariant to subjects' age and sex. In similar experiments, Birznieks et al (2001) reported that most of the changes to the finger's structure occur in the fingertip pulp, and not between the fingernail and the bone.

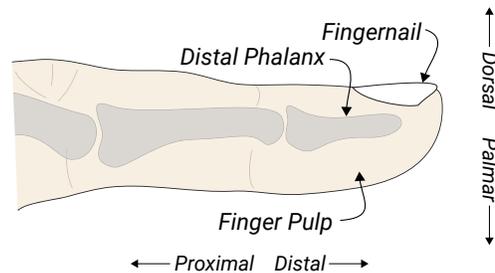
Sakai and Shimawaki (2006) examined a finger's contact area at acute angles and found that contact length (along the axis of the finger) increased non-linearly at force levels under 3 N, and saturated thereafter. The change in contact length between force levels was more pronounced as the angle of contact became more acute – caused by a difference in dorsal and proximal pulp compliance.

Goguy et al (2018a) characterised finger pitch, roll, and yaw in a series of fundamental touch operations (e.g. tapping, dragging, and flicking – see Section 2.3) with each finger and the thumb. They found touch operations generally occurred at a low pitch (less than  $45^\circ$ ), but with significant effects for the digit used and the orientation of the touch surface. Finger posture has also been studied in specific task contexts – for example, Azenkot and Zhai (2012) reported systematic shifts in touch distribution patterns with different typing patterns, which were later used to improve text entry performance (Yin et al, 2013; Goel et al, 2013).

Even when force is not a parameter to an action, touch gestures necessarily convey a certain force level and profile – particularly for gestures that involve extended motion. Taher et al (2014) analysed these inherent force profiles (but explicitly not the force level) for common interactions: tapping, typing, zooming, rotating, and panning. They found typing and panning were generally characterised by a sharp increase and decrease in force, with a slightly extended force plateau at the peak for tapping actions (hypothesised to be a confirmation phase). For gestures that involved extended interaction, force varied with the distance between the fingers (e.g. when zooming), and with substantial variation in the profiles that included use of the thumb.

### 2.3 Touch Interaction Design

Touch, by its nature, encourages a direct-manipulation style of interaction. The absence of disparity in space or time (i.e. immediacy) elicits direct finger actions



**Fig. 4** The primary parts of a fingertip.

(Lee et al, 1985). This is in contrast to desktop interaction, where a cursor separates an intended point of interest from an action invocation, and an input device separates the user's input from the movements of the cursor.

As a result, modern touch interactions adopt direct manipulation characterised by an instantaneous connection between the finger's action and its functional outcome: *tapping* an object activates or selects it; *dragging* or *sliding* over a distance moves an object, translates a view, or draws a stroke; and *pinching* or *spreading* adjusts a zoom factor.

An exception to direct touch interaction is the *touch and hold (long press)* gesture: a user first touches an object, and then holds their position for a predetermined time threshold (typically 400–500 ms). The temporal disparity from the time threshold creates a latency that separates the user's action from the system's response. That is, the system's response is triggered as a result of the user's *inaction*, rather than a parameter of their physical action. As reviewed below, this disparity has sometimes been remedied by using the force of the input as a triggering mechanism.

However, direct interaction with a finger is not without costs: the finger's size (in comparison to a cursor or a stylus tip) and its necessary occlusion of targets on a touch screen precludes very precise input – known as the *fat finger problem* (Vogel and Baudisch, 2007; Holz and Baudisch, 2010, 2011; Bi et al, 2013). There has been substantial research on improving these accuracy problems, although typically by decreasing the directness of the interaction through intermediary devices or widgets (e.g. Potter et al, 1988; Albinsson and Zhai, 2003; Vogel and Baudisch, 2007; Benko et al, 2006)

### 2.3.1 Force-Based Interactions

Presuming that the force of a contact can be reliably measured (either with dedicated force sensors or a synthesised approximation using other available sensors), researchers have experimented with interaction use cases such as continuous input controls for scrolling (Antoine et al, 2017; Baglioni et al, 2011; Miyaki and Rekimoto, 2009), zooming (Suzuki et al, 2018; Miyaki and Rekimoto, 2009), selecting between modes of operation (Brewster and Hughes, 2009), context menus (Wilson et al, 2010; Heo and Lee, 2011b; Goguey et al, 2018b), and gesture operations (Rendl et al, 2014; Rekimoto and Schwesig, 2006).

Researchers have also studied the benefits and limitations of 'pseudo-force' indicated by an overt 'rocking' or 'rolling' gesture (Benko et al, 2006; Forlines et al, 2007; Arif and Stuerzlinger, 2013; Boring et al, 2012; Heo and Lee, 2013).

Boceck et al (2019) used a neural network on individual touch sensor images to estimate static touch force. However, despite limiting their model to index-finger data at a fixed posture (with the device resting on a flat surface), their model suffered from substantial error variance.

### 2.3.2 Developing a Deeper Touch Interaction

The touch-and-hold gesture is the weakest of the common touch gestures due to its indirectness: it is difficult for users to discover or perform if the time threshold is too long, or prone to misclassification as a tap gesture if the threshold is too short. These problems are particularly acute on mobile devices where there is strong demand for providing a wide range of interactive functionality within the limited physical bounds of the display and input space. Force sensing offers a possible mechanism for creating a direct *press* gesture, but has been challenged by the practical difficulties of providing it in commercial hardware.

The biomechanical interaction between the finger and a touch-sensor reviewed in Section 2.2 suggests that there is an opportunity to use the dynamic properties of the finger to sense a more natural, direct *pressing* gesture. As touch sensor data does not inherently contain force information (Equation 1), it is not possible to quantify the force at which a user is pressing. However, the temporal changes in the touch sensor data due to the biomechanical effects should reveal whether the force is qualitatively changing. The remainder of this chapter describes the design and development of a deep learning approach to sensing this change to provide a force-based direct *press* gesture.

## 3 Deep Touch Model

*Deep touch* aims to discriminate a user's touch intentions based on an understanding of the biomechanical interface between their finger and a capacitive touch sensor, and its temporal evolution as a gesture is performed. In particular, we aim to sense a force-based *press* gesture to create a direct means of interaction.

Instead of estimating the parameters of tap location, force level, or scroll distance, deep touch is explicitly focused on the classification of a gesture (e.g. tapping, pressing, or scrolling). It takes a holistic and dynamic approach in processing the temporal changes in touch sensor data: rather than estimating a finger's force at each frame of data and applying a heuristic over the estimated force, the sequence of frames are considered together as to whether they represent increasing force. This is advantageous because a single touch sensor image of an arbitrary finger contact does not contain any force information, but through a sequence of such images the application of force can be observed.

We develop anticipated patterns of touch sensor signals from the reviewed biomechanical literature and apply them to specific touch-gesture designs. Each gesture is designed in terms of a finger's contact with a touch sensor: how that contact evolves over the course of the gesture and how that contact is realised in a temporal sequence of touch sensor images.

These patterns can be identified in data by a neural network that combines learned spatial features and learned temporal features. Rather than using an unconstrained deep neural network, we use the anticipated touch sensor signal patterns to limit the

size of the model so that it is possible to use the network for real-time inference within the constraints of a production environment (see Section 4).

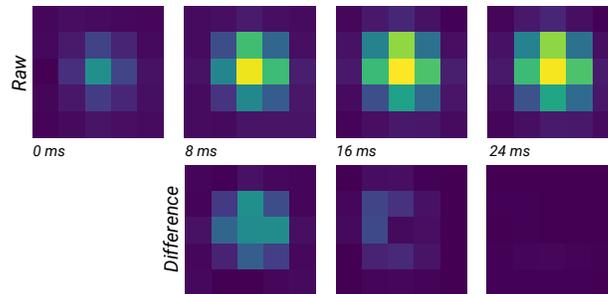
In the remainder of this section we detail these biomechanical patterns of touch sensor signals, and the design of the model. We then describe a data collection procedure for gathering labelled data to train the model, and the results of an offline evaluation.

### 3.1 Touch Gesture Patterns

Although we aim to sense a force-based *press* gesture with our model, it must be able to reliably discriminate this gesture from other touch gestures, namely *tapping* and *scrolling*. We therefore describe analyses for these three gestures in terms of the features that can be used to discriminate them.

A *tap* is conceptually the simplest touch interaction: a finger comes into symmetrical contact with a touch sensor, reaches a stable saturation point, and then disengages (lifts) from it. From the perspective of a touch sensor, the finger’s contact expands symmetrically around its centre of mass (Figure 5). There is little modulation of force after it saturates (Taher et al, 2014), and therefore the contact size or area will not further increase after the first few frames.

As a user evolves a contact into a *press* by applying more force behind their finger, the biomechanical literature informs us this will be conducive to an asymmetric contact expansion along the axis of the finger (Serina et al, 1997, 1998; Birznieks et al, 2001; Sakai and Shimawaki, 2006; Srinivasan et al, 1992; Pawluk and Howe, 1999). This will be prominent given that touch interactions generally occur at a low pitch (Goguey et al, 2018a). The touch sensor will therefore observe an expansion of the contact mass in one direction, while remaining ‘anchored’ at one edge (Figures 1 and 6).



**Fig. 5** A *tap* gesture observed by a touch sensor: the signal values (cell brightness) change symmetrically around the centre of mass. The top row shows the raw frames; the bottom row shows each frame’s differencing from the preceding frame.

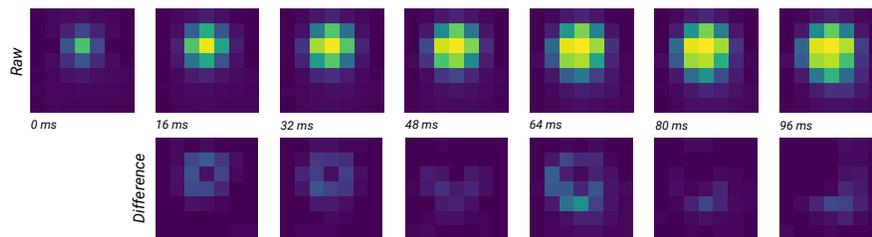
It is important to note that this assumes that either tapping gestures will be performed with a force of less than 1 or 2 N (i.e. before the contact area saturates – Serina et al, 1997, 1998; Sakai and Shimawaki, 2006), or the speed of the force onset will be significantly different during a tap. However, an advantage of this design is its invariance to the finger used to make the contact – that is, although a thumb and little finger will have substantially different contact areas, the relative changes as force is applied will be similar.

*Scrolling* interactions – both *dragging* and *flicking* (Quinn et al, 2013) – are primarily characterised by their contact displacement. This is facilitated in current systems by a ‘touch slop’ or hysteresis threshold to engage a scrolling mode (and exclude the possibility of tapping or pressing). Such a threshold is required because a contact point will rarely be stationary during tap and press interactions: jitter from the user’s muscle tremor and from the unfolding contact area will retard the centroid location (Wang et al, 2009; Wang and Ren, 2009).

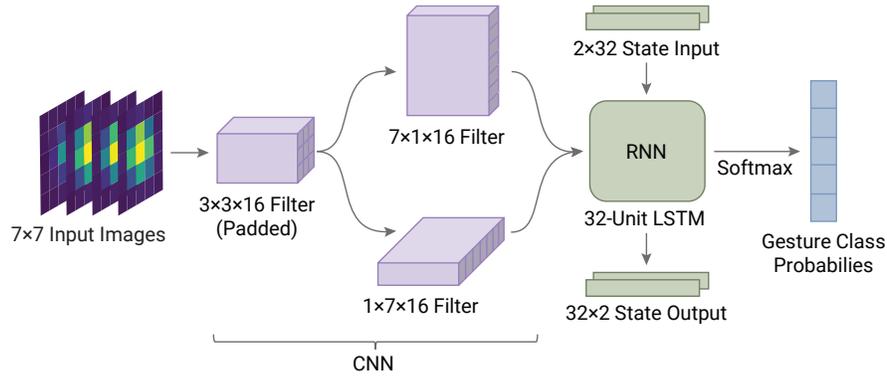
This displacement is conveyed in the touch sensor image through changes at the fringes of a touch. If the image is held at the calculated centroid, then motion is conveyed through a consistent decrease in signal at one edge, with a matched increase in signal at the other.

### 3.2 Model Design

Classification of touch gestures must occur online, in real-time, from continuous, variable-length time series data (i.e. without waiting for the finger to lift from the touch sensor). That is, the identification of a touch gesture should be made as soon as the user’s intent is sufficiently expressed – without further perceptible delay in time (in the case of *tapping* or *pressing*) or in space (in the case of *scrolling*). The identification also needs to be incremental, and not based on the entire gesture after its completion. Such a task lends itself to classification with a recurrent neural network (RNN) (e.g. Graves, 2012): touch sensor images can be input to the network as they are received, with the network’s state preserved between each image. The output



**Fig. 6** A *press* gesture observed by a touch sensor: the signal values (cell brightness) change asymmetrically around the centre of mass. The top row shows the raw frames; the bottom row shows each frame’s difference from the preceding frame.



**Fig. 7** An overview of the deep touch model’s design.

probability of each gesture class can be updated and compared against a threshold after each image is received – minimising classification latency. In particular, Long Short-Term Memory (LSTM) units allow for complex temporal patterns across the input to be identified (Hochreiter and Schmidhuber, 1997).

Our model was designed to capture a learned sequence of the axial features described above, and is illustrated in Figure 7. The input is a baselined  $7 \times 7$  (16-bit; single-channel) image from the touch sensor, cropped around the calculated centroid. This image is large enough to cover all reasonable touch contacts while minimising bandwidth requirements (i.e. image data can be transmitted with centroid data at 120 Hz – the native frequency of the touch sensor).

Each touch sensor image is first passed through a convolutional neural network (CNN). The image is convolved with a  $3 \times 3$  filter,<sup>3</sup> and is padded with zeros to produce a filtered image of the same height and width. This filtered image is processed separately by  $1 \times 7$  and  $7 \times 1$  filters to extract ‘row’ and ‘column’ features – reflecting the patterns of axial changes in contact mass and area. The concatenation of these features is fed into a recurrent layer (32 unit LSTM), which produces a gesture class output vector (softmax via linear activation) and a  $2 \times 32$  state vector. For a sequence of images, the state vector is initialised with zeros, and is preserved between them.

### 3.3 Data Set Development

In many domains (e.g. computer vision or natural language processing), samples of natural phenomena can be collected into a corpus and used to train a model. However, interactive gestures are artificial – they are constructed by a designer that is expecting users to perform a particular set of inputs – and therefore must be collected by eliciting them experimentally.

<sup>3</sup> All convolutional filters have a depth of 16, with ReLU activation between each operation (Glorot et al, 2011).

We collected a data set of labelled capacitive touch sensor image sequences that were representative of tap, press, and scroll operations on mobile devices in three tasks: (1) a target selection task, (2) a dragging task, and (3) a search-and-select task. The first two tasks asked for mechanical performances of common interactions, while the third was indicative of actual interaction and interleaved sequences of scrolling with tapping.

We collected data for both *long press* and *deep press* tasks separately: *long press* was defined using the system’s standard time-based threshold of 400 ms, while *deep press* was a user-defined force gesture. Similarly, we divided *scroll* tasks into *flicking* and *dragging* tasks: *flicking* tasks were scrolling actions of medium–long distances, while *dragging* tasks were micro-scrolling movements. The data for each of these tasks were labelled with their respective categories: *tap*, *deep press*, *long press*, *flick*, and *drag*. The reason for these divisions was to separate different finger motions (e.g. drag and flick) that generate the same touch gesture (e.g. scroll), and ensure that the principal features of the underlying motions can be identified by the model.

### 3.3.1 Participants & Apparatus

Nineteen volunteers (11 male; 8 female) with an age range of 18–60 participated in the experiment and received a gift voucher for doing so. The experiment was run on a Google Pixel 4 device with a  $144 \times 67$  mm display running at a resolution of  $3040 \times 1440$  px. The touch sensor had a resolution of  $36 \times 17$  cells, and reported a  $7 \times 7$  cell image centred on the cell that contained the calculated touch centroid at a rate of 120 Hz.

### 3.3.2 Task Design

We used three interaction tasks – *target selection*, *dragging*, and *search-and-select* – to elicit the five classes of gestures identified earlier.

The *target selection* task (Figure 8a) placed 12 circular targets of varying radii across the screen, and asked subjects to perform a particular touch gesture on each one (*tap*, *long press*, or *deep press*). Targets had their radius (105, 140, or 175 px), horizontal location (20, 50, or 80% of the screen width), and vertical location (9, 36, 64, or 91% of the screen height) randomly sampled to ensure interactions were distributed across the display. Only one target was visible at a time. Data from this task were labelled with the requested touch gesture.

Participants received haptic feedback when they were asked to long press (the system’s default behaviour). However, no feedback was given for deep-press tasks in order to avoid biasing participants towards a particular biomechanical performance – any interaction on a target was recorded as a *deep press* sample.

The *dragging* task (Figure 8b) simulated fine scrolling tasks (e.g. moving a cursor within a text field) by asking participants to drag a solid target to a hollow dock. The targets were created and displayed as with the target selection task, but with

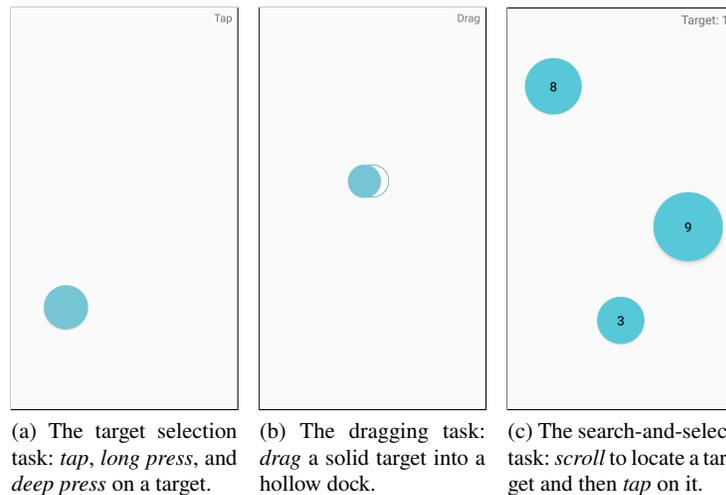
smaller target radii (70, 105, or 140 px) to simulate a fine-scrolling scenario like text selection. The dock was placed at varying distances (35, 53, or 70 px) and directions (up, down, left, right) from the target. Targets were initially blue, and turned orange with haptic feedback when they neared the dock – indicating that the task could be completed by lifting the finger. Data from this task were labelled as *drag* samples.

The *search-and-select* task (Figure 8c) included tapping and flicking actions. Each task required participants to *scroll* and locate a circular target (using a combination of dragging and flicking), and perform a *tap* on the target. The 12 targets were created and displayed by random sampling their radius (140, 175, or 210 px), horizontal location (20, 50, or 80% of the screen width), and vertical location (25, 38, or 63% of the screen height between targets) to ensure interactions were distributed across the display. Each target had a randomly assigned label between 1 and 12, and the task proceeded sequentially through them (cued to participants in the top-right corner of the display). Data from this task were labelled as *performed*.

### 3.3.3 Procedure

Participants were encouraged to perform a deep press using their preferred force and duration (potentially shorter than the current long-press duration), and to perform the tap, long press, drag, and flick operations as usual.

Each task (one *target selection* task for each gesture, one *dragging* task, and one *search-and-select* task) was performed in a counter-balanced order, and repeated as three blocks. The tasks were repeated in each of four postures (counter-balanced): (1) one-handed using a thumb to interact, (2) two-handed using either thumb to interact, (3) one-handed using the opposing index finger to interact, and (4) in a



**Fig. 8** The design of the data collection tasks.

landscape orientation with both hands, using either thumb to interact. In all postures the phone was hand-held. Participants could rest between blocks.

This procedure was repeated twice: once with a rubber case on the device, and once without a case. This ensured that touch sensor data were collected in both electrically grounded and ungrounded conditions – which have substantially different signal and noise characteristics.

### 3.4 Training

To train the model described above, the collected data were randomly divided into training (15 participants) and evaluation (4 participants) sets. No participant contributed samples to both sets. The model’s output was configured to estimate probabilities for five classes: *tap*, *deep press*, *long press*, *flick*, and *drag*.

To isolate the portion of each sample that contained the gesture performance, the trailing 25% of each sequence was discarded. This effectively removed the portion where the participant’s finger lifted from the touch sensor (i.e. after the gesture had been performed).

Each training sample was also extended to a minimum duration of 48 ms (6 frames) by linear interpolation, and truncated to a maximum duration of 120 ms (15 frames). This prevented certain touch gestures from being discriminable purely by their duration (in practice we expect to observe more variance in duration than captured in the laboratory). This processing was not applied to the samples used for evaluation.

We used the summed cross-entropy across each sequence as the loss function to minimise, with a linear temporal weight. That is, given a sequence of frames  $t \in [1, T]$  with a true class distribution at each frame  $p_t$ , and a predicted class distribution at each frame  $q_t$ , the loss over the classes  $\mathcal{X}$  was:

$$\mathcal{L}(p, q, T) = - \sum_{t=1}^T \left[ \frac{t}{T(1+T)/2} \sum_{x \in \mathcal{X}} p_t(x) \log q_t(x) \right].$$

As with earlier weighted cross-entropy methods, this encourages the model to produce classifications with an increasing probability for the correct class as input is received (Aliakbarian et al, 2017; Ma et al, 2016). However, in our formulation the weights always summed to 1 in order to make the total loss invariant to the length of the sequence, and avoid a potential bias in the model towards classes with shorter sequences.

To reflect the temporal ambiguity in the sequence, the true class distribution was defined at each frame with a logistic function. As the first few frames of a sequence for all classes are likely to be substantially similar (i.e. at the moments a finger first contacts the touch sensor), it is unreasonable to claim there is a high likelihood in the sequence’s ultimate classification for such frames (i.e. as with a one-hot encoded probability distribution). Similarly, it is unreasonable to penalise the model with

a high cross-entropy if it does not produce a confident prediction at these early frames. Therefore, the distribution  $p_t$  was defined to start at  $1/|\mathcal{X}|$  for all classes, and transition towards either 1 or 0, depending on the true label  $X$  for the sequence:

$$p_t(x) = \begin{cases} \frac{1}{(|\mathcal{X}| - 1)e^{-t+1} + 1} & x = X. \\ \frac{e^{-t+1}}{(|\mathcal{X}| - 1)e^{-t+1} + 1} & x \neq X. \end{cases} \quad (2)$$

Defining the true class distribution in this manner also helps calibrate the output probabilities and avoid spurious values in the first few frames during inference.

### 3.5 Results

To verify our patterns of axial changes in sensor images and demonstrate the importance of using temporal weights in the loss function, we conducted ablation studies with three model variations: (1) the complete model, as described above; (2) the model trained without the ‘row’ and ‘column’ convolutional filters, (3) the model trained without the temporal labels (Equation 2). Table 1 shows the overall accuracy and *deep press* precision/recall for these three models – with the removal of the row/column filters or the temporal weights having a substantial negative effect on the model’s performance.

Table 2 shows the confusion matrix for the evaluation data set, with an overall accuracy of 83%. When considering *deep press* as a binary class (i.e. *deep press* vs. *not-deep-press*), the overall accuracy is 95% with a precision of 89% and a recall of 75%.

In general, there is good separation between the classes, with the primary areas of confusion being between *deep press* and the *long press/drag* classes. However, a significant caveat with the reported *deep press* accuracy is the lack of feedback given to participants during the data collection procedure. The collected dataset was deliberately harder to classify than gestures with feedback will be in practice. That is, any action on the *deep press* targets were accepted and labelled as such, without constraint or validation. There are therefore likely to be poor samples in the data

**Table 1** Overall model accuracy and deep press precision/recall for the model component ablations.

	Accuracy	Deep Press	
		Precision	Recall
Complete model	83%	88%	78%
Without row/column filters	73%	69%	48%
Without temporal labels	76%	66%	67%

**Table 2** The confusion matrix for the offline model evaluation.

		Predicted				
		Tap	Deep	Long	Flick	Drag
Actual	Tap	98%	—	—	—	2%
	Deep	—	78%	14%	—	8%
	Long	—	9%	60%	—	31%
	Flick	1%	—	1%	95%	3%
	Drag	2%	—	12%	7%	79%

from either accidental touches or postures that do not produce a distinct ‘press’ (e.g. fingers approaching orthogonal to the display).

Creating a feedback loop would give users the opportunity to learn the distinguishing characteristics of the gesture, and drive them towards discriminating their own actions (Kaaresoja et al, 2014). This issue is addressed in the following section.

## 4 System Integration & Evaluation

The prior section demonstrated that temporal changes in touch sensor images convey distinct signals that can be used to discern a force-based *press* gesture. However, the deep touch model does not eliminate the need for heuristic classification of touch gestures as not all touch intentions involve predictable finger-based interactions (e.g. a conductive stylus or certain finger postures would not exhibit the biomechanical properties described). Rather, the model provides a method for *accelerating* the recognition of a user’s intentions when they are clear from the contact posture.

In this section we describe incorporating the deep touch model into the Android input system to enable its practical use (Quinn and Feng, 2020). This involves combining the neural deep touch model with the existing heuristic classification algorithm – allowing the neural model’s signals to accelerate classification when they become manifest in the touch sensor data, but falling back to traditional classification for unusual postures or when there is ambiguity in the signal. We then describe a user study to examine the practical classification performance of this algorithm.

### 4.1 Gesture Classification Algorithm

The Android input system provides signals about touch gestures to applications in the three categories discussed: *tap*, *press*, and *scroll*.<sup>4</sup> Applications typically map *press* signals to secondary invocation functions (e.g. context menus or word selection),

<sup>4</sup> <https://developer.android.com/reference/android/view/GestureDetector>

and so we decided to supplement this signal with our neural model classification – that is, allowing a *press* gesture to be triggered by either a *long press* or a *deep press*. This supports our goal of providing a direct touch gesture, but does not require existing applications to modify how they handle touch gestures in order to benefit from it.

Combining a probabilistic model with a heuristic algorithm offers two benefits to users: (1) lower latency for interactions when the intention is clear from the touch expression, and (2) the certainty and reliability of a baseline in the presence of input ambiguity. We therefore prefaced the existing gesture detection algorithm (steps 2–4 below) with a decision point for the neural model’s classification (Figure 9):

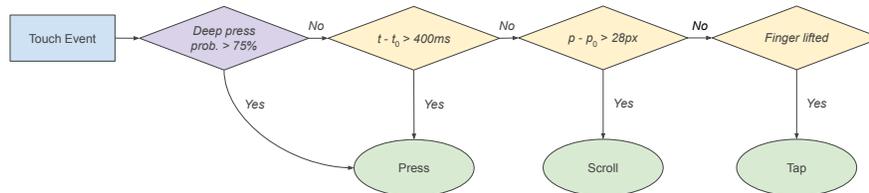
1. If the neural model indicates the sequence is a ‘deep press’ with a probability greater than 75%, the gesture is classified as a *press*. (However, if the neural model indicates similar confidence in another gesture classification, then only the heuristics are used for the remainder of the sequence.)
2. If the duration since the initial contact exceeds a time threshold of 400 ms, the gesture is a *press*.
3. If the distance from the initial contact location exceeds a hysteresis threshold, the gesture is a *scroll*.
4. Otherwise, the gesture is a *tap* when the finger contact is lifted.

Once a sequence has been classified, it is never reconsidered.

The hysteresis threshold for *scroll* classification was dynamically set based on the output of the model: 56 px while the neural model’s output was below the probability threshold of 75% for any of the gesture classes, and 28 px thereafter. That is, the threshold was doubled while the neural model expressed that the sequence was ambiguous – as it is common for some erroneous shift in the touch centroid as a finger’s area expands into the touch sensor during a press.

This algorithm allows the heuristic criteria to identify *tap* and *long-press* gestures when the model is unable to confirm an interaction as a *press*. However, the model’s training on five classes allows it to learn the discriminating characteristics of all possible interactions.

The neural model was implemented using TensorFlow Lite with an on-disk size of 167 kB, and a runtime memory load of less than 1 MB. When executing the model



**Fig. 9** An overview of the inference algorithm: the neural model is integrated into the heuristic classification pipeline to provide an acceleration for *press* gestures when the model’s output indicates high confidence.

on a Google Pixel 4 device, inference time averages  $50\mu\text{s}$  per input frame. This allows the model to execute for each image received from the touch sensor (i.e. at 120 Hz) and report its results to applications without impacting touch latency or system performance.

## 4.2 Evaluation

We conducted an evaluation of the algorithm to examine its performance. The evaluation followed a similar design and procedure as the data collection described earlier, but with two key differences: (1) haptic feedback was given to subjects when a *deep press* was detected (matching that for *long press*), and (2) the targets in the *search-and-select* task were either labelled *tap* or *press* (not just *tap*) to simulate realistic usage of the two touch expressions.

The gesture prompts to participants were the same as during data collection, and correctness was not enforced (e.g. a participant could perform a *tap* on a *press* target, which would be recorded as a failed classification).

Fourteen volunteers (10 male; 4 female) with an age range of 18–60 participated in the experiment and received a gift voucher. The experiment was run on the same type of device used for data collection. Due to time constraints, the use of a rubber phone case was a between-subjects condition.

Table 3 shows the confusion matrix across all data in the user study. As the gesture classification algorithm does not distinguish between the type of *press* – *long* or *deep* – we consider them together. This matches the experience that users will receive in practice, as the system response for all types of *press* – secondary invocation – is identical. The performance for binary *press* classification has a precision of 97% and a recall of 88%, with an average time to classify a *press* with the neural model of 235 ms (from the initial contact to the crossing of the probability threshold).

The largest source of confusion for *press* was with *scroll*. Much of this was due to a shift in the centroid caused by the expanding finger contact exceeding the hysteresis threshold before the model probability threshold. This occurred at approximately twice the rate for *deep press* than for *long press*, which is unsurprising given that *deep press* encourages an expanding contact area that may affect the centroid. The confusion between *press* and *tap* occurred at the same rate for *deep press* and *long press*, and was likely due to user error.

**Table 3** The confusion matrix for the online model evaluation.

		Predicted		
		Tap	Press	Scroll
Actual	Tap	99%	1%	—
	Press	4%	88%	8%
	Scroll	—	2%	98%

The rate of false-positive *scrolls* can be balanced against false-positive *presses* using the scroll hysteresis threshold, which is weighted by the cost of different types of classification errors. For instance, a false-positive press is likely to be more costly to a user than a false-positive scroll because a press typically invokes some action that may be difficult for a user to reverse or correct, whereas a scroll may only displace the content.

## 5 Discussion

The deep touch neural model uses the biomechanical signals captured by a touch sensor to identify force-based *press* gestures from users without dedicated force-sensing hardware. By extracting spatial and temporal features in the touch image sequence, the deep touch neural network can enhance the modern touchscreen gesture experience beyond what conventional heuristics-based gesture classification algorithms could do alone. The model can be executed in a production environment (delivered with Google Pixel 4 and Pixel 5 devices) without increasing touch input latency or impairing system performance.

Instead of creating a new interaction modality, we focused on improving the user experience of long press interactions by accelerating them with force-induced deep press in a unified *press* gesture. A press gesture has the same outcome as a long press gesture, whose time threshold remains effective, but provides a stronger connection between the outcome and the user's action when force is used. This allowed us to create a more natural and direct gesture to supplement the conventional, indirect *touch and hold* gesture.

Combining a neural model with the existing heuristic method of gesture detection allows biomechanical information to be identified and utilised when it is present, but without harming the usability of touch input for other finger postures. However, this means that the relationship between the heuristic criteria and the probabilistic output of a neural model needs to be carefully considered. Specifically, in cases of ambiguity the system may want to err towards the least costly or most consistent classification for the user, rather than the most accurate.

This is most visible in the confusion between a *press* and a *scroll*, where the expanding contact area of the *press* gesture erroneously induces a change in the touch centroid that triggers a heuristic *scroll* classification. There are further opportunities here to either tune the scroll hysteresis threshold, or to leverage the neural model to aid in classification of a *scroll* gesture as well.

While data curation and training are key to any successful neural network development, they are particularly important and challenging in solving low level HCI problems with neural networks where the human actions and their effects and feedback are linked in a tight interaction loop. Lacking naturally existing datasets that can be labeled offline, we took a data elicitation approach in developing the deep touch model, by asking human participants to intuitively perform touch gestures as they expect and against a set of tasks. However, this data collection procedure for training

samples lacked haptic feedback for the deep press gesture, which might have affected its offline classification performance (Table 2). Potentially the training datasets and the network’s performance can be further enhanced by a closed-loop data collection with haptic feedback for all touch gestures, with the feedback driven by the current deep touch model.

Neural models are also well-suited for touch interactions beyond those studied here – and the human–computer interaction literature has many examples. For example, finger rolling (Roudaut et al, 2009), ‘pushing’ and ‘pulling’ shear forces (Heo and Lee, 2011a; Harrison and Hudson, 2012; Heo and Lee, 2013; Lee et al, 2012), and ‘positive’ and ‘negative’ force gestures (Rekimoto and Schwesig, 2006) might be supported with similar biomechanical patterns. This style of analysis may also provide insight into perceived input location issues (Holz and Baudisch, 2010) and improved touch contact location algorithms by capturing more information about the contact mass.

## 6 Conclusion

This work demonstrates that combining capacitive touch sensing with modern neural network algorithms is a practical direction to improve the usability and expressivity of touch-based user interfaces. The work was motivated by a *deep touch* hypothesis that (1) the human finger performs richer expressions on a touch surface than simple pointing; (2) such expressions are manifest in touch sensor image sequences due to finger–surface biomechanics; and (3) modern neural networks are capable of discriminating touch gestures using these sequences. In particular, a deep *press* gesture, accelerated from long press based on an increase in a finger’s force could be sensed by a neural model in real time without additional hardware, and reliably discriminated from tap and scroll gestures. The *press* classification has a precision of 97% and a recall of 88%, with an average time reduced to 235 ms from the conventional 400–500 ms) long press.

More broadly, input sensors often capture rich streams of high-dimensionality data that are typically summarised to a few key metrics to simplify the development of heuristic analyses and classifications. Neural methods permit the analysis of the raw data stream to find more complex relationships than can be feasibly expressed with heuristics, and computational advances have made it feasible to operationalise these models in real-time. This chapter has described a practical instance of this – *deep touch* – where a neural model has enhanced existing heuristic methods, and been deployed widely to enable a richer user experience.

## Acknowledgement

We thank many Google and Android colleagues in engineering, design and PM for their direct and indirect contributions to the project. The work would have not been possible without the machine learning and framework infrastructure they have built.

## References

- Albinsson PA, Zhai S (2003) High precision touch screen interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, CHI '03, pp 105–112, DOI 10.1145/642611.642631
- Aliakbarian MS, Saleh FS, Salzmann M, Fernando B, Petersson L, Andersson L (2017) Encouraging LSTMs to anticipate actions very early. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 280–289, DOI 10.1109/ICCV.2017.39
- Antoine A, Malacria S, Casiez G (2017) Forceedge: Controlling autoscroll on both desktop and mobile computers using the force. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '17, pp 3281–3292, DOI 10.1145/3025453.3025605
- Arif AS, Stuerzlinger W (2013) Pseudo-pressure detection and its use in predictive text entry on touchscreens. In: Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, ACM, New York, NY, OzCHI '13, pp 383–392, DOI 10.1145/2541016.2541024
- Azenkot S, Zhai S (2012) Touch behavior with different postures on soft smartphone keyboards. In: Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, ACM, New York, NY, MobileHCI '12, pp 251–260, DOI 10.1145/2371574.2371612
- Baglioni M, Malacria S, Lecolinet E, Guiard Y (2011) Flick-and-Brake: Finger control over inertial/sustained scroll motion. In: CHI '11 Extended Abstracts on Human Factors in Computing Systems, ACM, New York, NY, CHI EA '11, pp 2281–2286, DOI 10.1145/1979742.1979853
- Barrett G, Omote R (2010) Projected-capacitive touch technology. *Information Display* 26(3):16–21
- Benko H, Wilson AD, Baudisch P (2006) Precise selection techniques for multi-touch screens. In: CHI '06, ACM, New York, NY, pp 1263–1272, DOI 10.1145/1124772.1124963
- Bi X, Li Y, Zhai S (2013) FFitts law: Modeling finger touch with Fitts' law. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '13, pp 1363–1372, DOI 10.1145/2470654.2466180

- Birznieks I, Jenmalm P, Goodwin AW, Johansson RS (2001) Encoding of direction of fingertip forces by human tactile afferents. *Journal of Neuroscience* 21(20):8222–8237, DOI 10.1523/jneurosci.21-20-08222.2001
- Boceck T, Le HV, Sprott S, Mayer S (2019) Force touch detection on capacitive sensors using deep neural networks. *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* DOI 10.1145/3338286.3344389
- Boring S, Ledo D, Chen XA, Marquardt N, Tang A, Greenberg S (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*, ACM, New York, NY, MobileHCI ’12, pp 39–48, DOI 10.1145/2371574.2371582
- Brewster SA, Hughes M (2009) Pressure-based text entry for mobile devices. In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM, New York, NY, MobileHCI ’09, pp 9:1–9:4, DOI 10.1145/1613858.1613870
- Buxton W (1995) Touch, gesture, and marking. In: Baecker RM, Grudin J, Buxton W, Greenberg S (eds) *Human-Computer Interaction: Toward the Year 2000*, Morgan Kaufmann Publishers, San Francisco, CA, chap 7, pp 469–482
- Buxton W, Hill R, Rowley P (1985) Issues and techniques in touch-sensitive tablet input. In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, SIGGRAPH ’85, pp 215–224, DOI 10.1145/325334.325239, URL <http://doi.acm.org/10.1145/325334.325239>
- Forlines C, Wigdor D, Shen C, Balakrishnan R (2007) Direct-touch vs. mouse input for tabletop displays. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Association for Computing Machinery, New York, NY, CHI ’07, pp 647–656, DOI 10.1145/1240624.1240726
- Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Gordon G, Dunson D, Dudík M (eds) *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, PMLR, Fort Lauderdale, FL, *Proceedings of Machine Learning Research*, vol 15, pp 315–323
- Goel M, Jansen A, Mandel T, Patel SN, Wobbrock JO (2013) ContextType: Using hand posture information to improve mobile touch screen text entr. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI ’13, pp 2795–2798, DOI 10.1145/2470654.2481386
- Goguy A, Casiez G, Vogel D, Gutwin C (2018a) Characterizing finger pitch and roll orientation during atomic touch actions. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI ’18, pp 589:1–589:12, DOI 10.1145/3173574.3174163
- Goguy A, Malacria S, Gutwin C (2018b) Improving discoverability and expert performance in force-sensitive text selection for touch devices with mode gauges. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI ’18, DOI 10.1145/3173574.3174051
- Graves A (2012) *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer-Verlag, Berlin Heidelberg, DOI 10.1007/978-3-642-24797-2

- Grosse-Puppenthal T, Holz C, Cohn G, Wimmer R, Bechtold O, Hodges S, Reynolds MS, Smith JR (2017) Finding common ground: A survey of capacitive sensing in human-computer interaction. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '17, pp 3293–3315, DOI 10.1145/3025453.3025808
- Harrison C, Hudson S (2012) Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '12, pp 3149–3152, DOI 10.1145/2207676.2208730
- Heo S, Lee G (2011a) Force gestures: augmenting touch screen gestures with normal and tangential forces. In: Proceedings of the 24th annual ACM symposium on User interface software and technology, ACM, New York, NY, UIST '11, pp 621–626, DOI 10.1145/2047196.2047278
- Heo S, Lee G (2011b) ForceTap: Extending the input vocabulary of mobile touch screens by adding tap gestures. In: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, ACM, New York, NY, MobileHCI '11, pp 113–122, DOI 10.1145/2037373.2037393
- Heo S, Lee G (2013) Indirect shear force estimation for multi-point shear force operations. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '13, pp 281–284, DOI 10.1145/2470654.2470693
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Computation* 9(8):1735–1780, DOI 10.1162/neco.1997.9.8.1735
- Holz C, Baudisch P (2010) The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In: Proceedings of the 28th international conference on Human factors in computing systems, ACM, New York, NY, CHI '10, pp 581–590, DOI 10.1145/1753326.1753413
- Holz C, Baudisch P (2011) Understanding touch. In: Proceedings of the 2011 annual conference on Human factors in computing systems, ACM, New York, NY, CHI '11, pp 2501–2510, DOI 10.1145/1978942.1979308
- Hu Y, Huang L, Rieutort-Louis W, Sanz-Robinson J, Wagner S, Sturm JC, Verma N (2014) 3D gesture-sensing system for interactive displays based on extended-range capacitive sensing. In: 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers, ISSCC, pp 212–213, DOI 10.1109/ISSCC.2014.6757404
- Kaaresoja T, Brewster S, Lantz V (2014) Towards the temporally perfect virtual button: Touch-feedback simultaneity and perceived quality in mobile touchscreen press interactions. *ACM Transactions on Applied Perception* 11(2):9:1–9:25, DOI 10.1145/2611387
- Lee B, Lee H, Lim SC, Lee H, Han S, Park J (2012) Evaluation of human tangential force input performance. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '12, pp 3121–3130, DOI 10.1145/2207676.2208727

- Lee J, Cole MT, Lai JCS, Nathan A (2014) An analysis of electrode patterns in capacitive touch screen panels. *Journal of Display Technology* 10(5):362–366, DOI 10.1109/JDT.2014.2303980
- Lee S, Buxton W, Smith KC (1985) A multi-touch three dimensional touch-sensitive tablet. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI '85, pp 21–25, DOI 10.1145/317456.317461
- Ma S, Sigal L, Sclaroff S (2016) Learning activity progression in LSTMs for activity detection and early detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1942–1950, DOI 10.1109/CVPR.2016.214
- Miyaki T, Rekimoto J (2009) GraspZoom: Zooming and scrolling control model for single-handed mobile interaction. In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM, New York, NY, MobileHCI '09, pp 11:1–11:4, DOI 10.1145/1613858.1613872
- Miyata N, Yamaguchi K, Maeda Y (2007) Measuring and modeling active maximum fingertip forces of a human index finger. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 2156–2161, DOI 10.1109/IROS.2007.4399243
- O'Connor T (2010) mTouch projected capacitive touch screen sensing theory of operation. Tech. Rep. TB3064, Microchip Technology Inc.
- Pawluk DTV, Howe RD (1999) Dynamic contact of the human fingerpad against a flat surface. *Journal of Biomechanical Engineering* 121(6):605–611, DOI 10.1115/1.2800860
- Potter RL, Weldon LJ, Shneiderman B (1988) Improving the accuracy of touch screens: an experimental evaluation of three strategies. In: *CHI '88*, ACM, New York, NY, pp 27–32, DOI 10.1145/57167.57171
- Quinn P, Feng W (2020) Sensing force-based gestures on the Pixel 4. Google AI Blog URL <https://ai.googleblog.com/2020/06/sensing-force-based-gestures-on-pixel-4.html>
- Quinn P, Malacria S, Cockburn A (2013) Touch scrolling transfer functions. In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, UIST '13, pp 61–70, DOI 10.1145/2501988.2501995
- Rekimoto J, Schwesig C (2006) PreSenseII: Bi-directional touch and pressure sensing interactions with tactile feedback. In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, ACM, New York, NY, CHI EA '06, pp 1253–1258, DOI 10.1145/1125451.1125685
- Rendl C, Greindl P, Probst K, Behrens M, Haller M (2014) Presstures: Exploring pressure-sensitive multi-touch gestures on trackpads. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI '14, pp 431–434, DOI 10.1145/2556288.2557146
- Rosenberg I, Perlin K (2009) The unmousepad: An interpolating multi-touch force-sensing input pad. *ACM Transactions on Graphics* 28(3):65:1–65:9, DOI 10.1145/1531326.1531371
- Roudaut A, Lecolinet E, Guiard Y (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, ACM, New York, NY, CHI '09, pp 927–936, DOI 10.1145/1518701.1518843
- Sakai N, Shimawaki S (2006) Mechanical responses and physical factors of the fingertip pulp. *Applied Bionics and Biomechanics* 3(4):273–278, DOI 10.1533/abbi.2006.0046
- Serina ER, Mote CD Jr, Rempel D (1997) Force response of the fingertip pulp to repeated compression—effects of loading rate, loading angle and anthropometry. *Journal of Biomechanics* 30(10):1035–1040, DOI 10.1016/S0021-9290(97)00065-1
- Serina ER, Mockensturm E, Mote CD Jr, Rempel D (1998) A structural model of the forced compression of the fingertip pulp. *Journal of Biomechanics* 31(7):639–646, DOI 10.1016/S0021-9290(98)00067-0
- Srinivasan MA, LaMotte RH (1995) Tactual discrimination of softness. *Journal of Neurophysiology* 73(1):88–101, DOI 10.1152/jn.1995.73.1.88
- Srinivasan MA, Gulati RJ, Dandekar K (1992) In vivo compressibility of the human fingertip. *Advances in Bioengineering* 22:573–576
- Suzuki K, Sakamoto R, Sakamoto D, Ono T (2018) Pressure-sensitive zooming-out interfaces for one-handed mobile interaction. In: *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM, New York, NY, MobileHCI '18, pp 30:1–30:8, DOI 10.1145/3229434.3229446
- Taher F, Alexander J, Hardy J, Velloso E (2014) An empirical characterization of touch-gesture input-force on mobile devices. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ACM, New York, NY, ITS '14, pp 195–204, DOI 10.1145/2669485.2669515
- Vogel D, Baudisch P (2007) Shift: A technique for operating pen-based interfaces using touches. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI '07, pp 657–666, DOI 10.1145/1240624.1240727
- Walker G (2012) A review of technologies for sensing contact location on the surface of a display. *Journal of the Society for Information Display* 20(8):413–440, DOI 10.1002/jsid.100
- Wang F, Ren X (2009) Empirical evaluation for finger input properties in multi-touch interaction. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, CHI '09, pp 1063–1072, DOI 10.1145/1518701.1518864
- Wang F, Cao X, Ren X, Irani P (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*, ACM, New York, NY, UIST '09, pp 23–32, DOI 10.1145/1622176.1622182
- Wang T, Blankenship T (2011) Projected-capacitive touch systems from the controller point of view. *Information Display* 27(3):8–11
- Westerman W (1999) Hand tracking, finger identification, and chordic manipulation on a multi-touch surface. PhD thesis, University of Delaware

- Wilson G, Stewart C, Brewster SA (2010) Pressure-based menu selection for mobile devices. In: Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services, ACM, New York, NY, MobileHCI '10, pp 181–190, DOI 10.1145/1851600.1851631
- Yaniger SI (1991) Force sensing resistors: A review of the technology. In: Electro International, pp 666–668, DOI 10.1109/ELECTR.1991.718294
- Yin Y, Ouyang TY, Partridge K, Zhai S (2013) Making touchscreen keyboards adaptive to keys, hand postures, and individuals: A hierarchical spatial backoff model approach. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, CHI '13, pp 2775–2784, DOI 10.1145/2470654.2481384
- Zimmerman TG, Smith JR, Paradiso JA, Allport D, Gershenfeld N (1995) Applying electric field sensing to human-computer interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., New York, NY, CHI '95, pp 280–287, DOI 10.1145/223904.223940