ShadowSense: Detecting Human Touch in a Social Robot Using Shadow Image Classification

YUHAN HU, Cornell University, USA
SARA MARIA BEJARANO, Cornell University, USA and Universidad de los Andes, Colombia
GUY HOFFMAN, Cornell University, USA

This paper proposes and evaluates the use of image classification for detailed, full-body human-robot tactile interaction. A camera positioned below a translucent robot skin captures shadows generated from human touch and infers social gestures from the captured images. This approach enables rich tactile interaction with robots without the need for the sensor arrays used in traditional social robot tactile skins. It also supports the use of touch interaction with non-rigid robots, achieves high-resolution sensing for robots with different sizes and shape of surfaces, and removes the requirement of direct contact with the robot. We demonstrate the idea with an inflatable robot and a standing-alone testing device, an algorithm for recognizing touch gestures from shadows that uses Densely Connected Convolutional Networks, and an algorithm for tracking positions of touch and hovering shadows. Our experiments show that the system can distinguish between six touch gestures under three lighting conditions with 87.5 – 96.0% accuracy, depending on the lighting, and can accurately track touch positions as well as infer motion activities in realistic interaction conditions. Additional applications for this method include interactive screens on inflatable robots and privacy-maintaining robots for the home.

CCS Concepts: • **Hardware** → **Tactile and hand-based interfaces**; • **Computer systems organization** → *Robotics*; • **Human-centered computing** → Human computer interaction (HCI); • **Computing methodologies** → Computer vision problems; Supervised learning by classification.

Additional Key Words and Phrases: Human-robot interaction; Tactile interaction; Image classification; Shadows

ACM Reference Format:
Yuhan Hu, Sara Maria Bejarano, and Guy Hoffman. 2020. ShadowSense: Detecting Human Touch in a Social Robot Using Shadow Image Classification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 132 (December 2020), 24 pages. https://doi.org/10.1145/3432202

1 INTRODUCTION

Touch is a highly salient channel of human communication and has also been acknowledged as an important modality for human-robot interaction (HRI) [35]. The role that the sense of touch plays in humans and animals have been widely studied, finding ties to emotional communication, attachment, bonding, intimacy and stress [11]. Hence, embedding a sense of touch in a social robot promises to be a natural way of interaction that can improve the robot's awareness of users' intentions, and promote intelligent and context-sensitive behaviors in HRI [4].

Despite the importance of touch interaction and the potential for rich tactile communication between humans and robots, very few social robots to date have integrated touch as a fine-grained interaction modality. One of the possible reasons for this gap is that existing tactile robot skins use force or capacitive sensors as the underlying

Authors' addresses: Yuhan Hu, yh758@cornell.edu, Cornell University, Ithaca, NY, USA, 14853; Sara Maria Bejarano, sm.bejarano44@uniandes.edu.co, Cornell University, Ithaca, NY, USA, 14853, Universidad de los Andes, Bogota, Cundinamarca, Colombia, 111711; Guy Hoffman, hoffman@cornell.edu, Cornell University, Ithaca, NY, USA, 14853.

2474-9567/2020/12-ART132
https://doi.org/10.1145/3432202

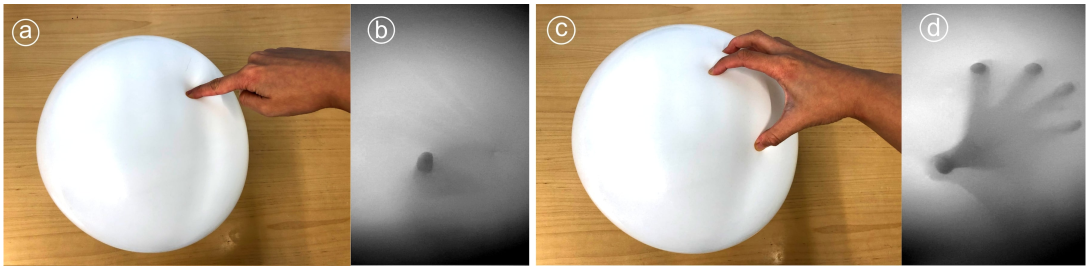

Fig. 1. ShadowSense can turn an inflatable skin into a touch-sensitive device by using a camera and image classification. (a) and (c) show two tactile gestures; (b) and (d) show the device's view of the shadows generated by these gestures.

sensor technology. One issue with these sensors is that they require a large number of units even for a relatively coarse spatial resolution. For example, the first MIT Huggable robot [36] integrated more than 1400 sensors to achieve full body detection. Subsequent versions of the robot used less sensors [15], but resulted in a low spatial resolution. Another shortcoming is that force sensitive resistors require a relatively flat and stiff surface, scale poorly to continuous coverage, and are insensitive to light touches, including those that interact with the robot above the skin surface. Capacitive sensors are less affected by curved surface and potentially more sensitive to light touches, but they are vulnerable to interference and more expensive than the proposed solution. Finally, sensor arrays require a prohibitively large amount of wiring or multiplexing to achieve a usable spatial resolution for touch interaction.

1.1 ShadowSense

Given this existing gap in HRI, our goal is to develop a low-tech, low-cost touch sensing alternative to recognize touch gestures and touch positions on social robots in the hope of improving the robot's awareness and promoting intelligent interaction behaviors. In this paper, we propose to use computer vision for tactile sensing, a method we call *ShadowSense* (Figure 1). A camera inside a robot's body captures the shadows created by touching its skin, or even hovering over the skin, detecting both touch positions and social gestures. This approach turns robots with translucent skins into touch-sensitive agents with the simple addition of a camera and computer vision software.

A similar approach has been used in the past for the inverse problem, namely to enable tactile sensors on robot grippers. Optical sensors on robot fingers were shown to assist robot grasp with the measurement of contact position, shape, and force estimation [22, 45]. However, such sensors have been limited to finger-tip scale, and have never been used to detect human touch in social robots.

ShadowSense was developed as part of our work on soft social robots. Soft and inflatable robots have garnered increasing attention as they can provide more adaptive and resilient movement, as well as a safer, more compliant interface with the environment. Soft and inflatable robots also offer new design opportunities for human-robot interaction [3, 32] and, given their soft exterior as affordances, are particularly suited for touch-based interaction. That said, the use of force sensors for tactile interaction makes this design choice infeasible, as soft robots generally undergo skin deformation when they are actuated and usually do not have a rigid skin even after inflation. From an interaction standpoint, the stiff sensors also interfere with the softness of the surface, and thus act against a satisfactory tactile user experience. This makes the use of ShadowSense especially apt for these kinds of robots.

1.2 Overview

In this paper, we demonstrate a proof-of-concept of an inflatable robot equipped with a camera, a neural-network-based algorithm to recognize touch gestures from shadow images, and a method to tracks hover shadows and touch positions. After pre-processing the images to overcome illumination bias, we use a Densely Connected Convolutional Neural Network ("DenseNet") and transfer learning to classify the shadow images. We then use a skin color segmentation-based algorithm to track the positions of hovering shadows and touch areas in the images. A state machine uses these tracks to determine the occurrence and direction of gestures.

In our experiments, we evaluated six interaction gestures: touching with a palm, punching, touching with two hands, hugging, pointing, and not touching. The classifier was evaluated with test images collected under three lighting conditions: daylight, dusk, and night. The results showed high accuracy in gesture recognition, between 87.5% and 96.0%, dependent on the lighting. We then evaluated the generalization of the ShadowSense classification by testing hold-out test sets of users and lighting conditions. The results indicated that the learned model generalizes well to unseen users with a mean accuracy of 87.9%, but does not generalize as well to unseen lighting conditions.

We evaluated the tracking algorithm using images of hover and touch gestures performed with a palm, a fist, and a fingertip. The proposed method is able to track the positions of touch and shadow gestures at a mean error of below 0.55 centimeters. We further demonstrate the performance of identifying motion activity from video sequences, which achieves more than 90% accuracy after observing only the first 50% of a full gesture. This suggests that ShadowSense is a promising approach for full-body human-robot interaction.

The combination of accurate touch localization and the ability to distinguish between a number of touch types can open the way for other interaction possibilities with robots, beyond social touch. The addition of a projector, for example, can give inflatable robots a touch-screen-like interface. An additional application of ShadowSense is to provide a privacy-conserving interaction method for home robots and other smart home devices.

2 RELATED WORK

Our work relates to three bodies of existing work. We first review literature about sensing methods used in tactile human-robot interactions. We then give a brief introduction of tactile sensors using computer vision methods. We finally discuss other kinds of hand-sensing human-computer interactions outside of robotics.

2.1 Tactile human-robot Interaction

To enable touch sensations on social robots, the most common techniques are force sensitive resistors (FSRs), capacitive sensors, electrical field sensors, and deformation sensors. For example, the Haptic Creature robot [4] uses a network of 56 surface-mounted FSRs to recognize touch, resulting in a correct classification rate of 36% on nine emotional expressions. The Huggable robot [36] utilizes over 1000 Quantum Tunnelling Composite (QTC) sensors, 400 temperature sensors, and 45 electric field sensing electrodes to achieve a high spatial resolution for detecting the social and affective content of touch. The robot Maggie [31] integrates a dozen capacitive sensors that are able to detect just contact; Robovie [14] uses 276 piezoelectric sensors; and CB2 [26] uses 197 piezoelectric sensors. Some social robots use less sensors to detect touch, for example, a huggable social robot developed by Sooyeon et al. [15] uses only 12 capacitive sensors and 2 pressure sensors. However, such designs result in a low spatial resolution.

These works share some common shortcomings, mainly the need to equip the robot with a large amount of sensors to cover the body. This is due to the fact that each sensor has a short detection range of a just few centimeters. Additionally, direct contact is required for most sensors and for some of the sensors, such as FSRs, light contact is barely detectable. Finally, implementing these types of tactile sensors becomes less feasible when the skin is highly nonplanar or made from a soft, deformable material.

A few other sensing methods have been explored. Acoustic sensing uses microphones to detect and classify touch gestures. For example, Fernando et al. [2] present a system that can distinguish between different types of touches on robot's shell using contact microphones. The classification resulted in relatively high accuracy recognizing four gestures: stroke, tickle, tap, and slap. SurfaceVibe [28] presents a vibration-based interaction-tracking system that allows for detection and tracking of two interaction gestures: tap and swipe, with localization error at a centimeter level. The acoustic sensing method results in low spatial resolution, applies only to solid surface and runs the risk of interference by external vibrations.

Global measurement of air pressure has also been proposed to provide force sensing on soft robots. Alexander et al. [3] present an air-filled force sensing module that can sense only the magnitude of contact force over a large area of the inflatable robot. However, only touches that apply force can be detected, and spatial information, such as the precise location of the touch, are hard to come by using this method.

Using a very different approach, Silvera et al. [35, 40, 41] presented a large-scale flexible sensitive artificial skin for robots using electrical impedance tomography (EIT). They carried out several experiments regarding touch interaction, and demonstrated its ability to recognize social touches at better-than-chance levels accuracy, based on three attributes: location, duration, and intensity of touch. Yet another approach is used in the PARO therapeutic robot [34, 42]. There, a dielectric material underneath the skin layer is able to sense the force magnitude and location. However, using such sensors also results in low spatial resolution and a resulting difficulty to detect and localize a large variety of touch gestures.

These works suggest a missing method for low-complexity, high-resolution tactile sensing for social robots.

2.2 Vision-based Tactile Sensors

Image sensors and computer vision methods offer an alternative to the above-mentioned methods of contact detection and classification, in particular with the promise of a higher sensing resolution. Indeed, several robotics researchers have used computer vision to detect objects being touched by a robot manipulator. The GelSight sensor [16, 17] is designed to measure the fine geometry profile of the contact surface. It is made of soft silicone gel, three color LEDs and a camera. The deformation of the gel during contact is captured by the camera on top of the gel. Li et al. [22] describe a fingertip GelSight sensor that can synthesize high-resolution height maps of object surfaces, and accurately determine the pose of a part grasped in robot's hand. They illustrate the capability to localize parts during the grasp process, and demonstrate the practicality in the context of a small parts insertion problem. Dong et al. [8] further improved the sensor by adding markers on the gel surface so that the sensor can detect contact areas and forces more accurately.

The TacTip sensor [43, 45] measures deformation of the sensing surface via the tracking of optical pins, a mechanism that mimics the intermediate ridge in the human skin. The performance of sensors is shown to attain submillimeter accuracy, and the technique shows promise to have real-world applications in tactile perception, manipulation, and exploration. Kappassov et al. [18] developed a color-coded optical tactile sensing array, incorporating plastic optical fibers inside silicone rubber. When the skin gets compressed after contact with an object, the scattering pattern of the light in the optical fibers changes, thus allowing for the measurement of force and the localization of the object. Huang et al. [13] used a small depth sensor on robot's fingertip to directly image the structure of the soft contact region, as well as a rough reconstruction of the stress induced by geometric material strain.

All of these sensors are limited to fingertip-scale sensing, and are intended to help robots understand the properties of objects they interact with, thus providing action related information, such as object localization and slippage detection. Their design mainly focused on force measurement, surface characteristics, and contact region estimation, as defined by the requirements of object perception for grasping. Detecting human touch on a whole-body scale in social robots, on the other hand, requires the capability to recognize a large variety of large touch gestures that can vary between users and is less concerned with measuring accurate force or fine geometries. Existing approaches to visual-based robotic touch sensors, having a high spatial resolution in small-scale contact force estimation and localization, are therefore not appropriate for the requirements of the social touch scenario.

Beyond directly capturing contact images with vision sensors, some researchers transform contact data acquired from tactile sensor arrays into an pseudo-image, and use image classification methods to process the data. Liu et al. [23] map each force measurement of a tactile sensor in a large-scale array to a pixel in a virtual contact image. They then used neural networks for object classifications on the contact images generated by the pressure sensors on a robot hand. Alessandro et al. [1] performed a human touch sensing experiment on the skin of a Baxter robot by transforming data acquired from 768 pressure sensors on the arm, and obtained a classification accuracy higher than 96%. These methods, though, still require massive amounts of sensors, as discussed in the previous section.

2.3 Sensing Hands in Human-Computer Interaction

In the context of user interface design, vision-based techniques have been used in human-computer interaction systems that used bare hands and hand gestures as input sources. In many cases, such systems apply computer vision techniques to detect and track finger locations or recognize hand gestures via overhead-mounted cameras. For example, Letessier et al. [21] present a finger tracking system with an overhead camera, detecting finger positions for direct manipulation of digital objects. Chung et al. [6] improve the system by using three side-mounted cameras that allow for more accurate finger capture. Some systems, e.g., Chiu et al. [5], use neural networks to classify touch gestures on the display surface, achieving 3.5% gesture error. Hand posture sensing is also used to develop touch screen surfaces for interaction. Z-touch [38] is a multi-touch table that senses users' hands and fingers postures in manipulating digital graphics, demonstrating applications such as drawing, map zooming, and curve control. TouchLight [44] presents a transparent imaging screen that allows users to interact with the image projection through gestures like dragging, tapping and sliding. Dohse et al. [7] presents a hand-tracking table-top that enables multiple users to collaborate and interact with projections on the table-top display. Schoning et al. [33] presents an optical diffusing soft surface for multi-touch tracking. Such systems focus on interaction with virtual displays on a 2-dimensional surface. The purposes of detecting hand poses are mainly for manipulating or interacting with virtual objects, such as selecting, drawing, and dragging a virtual shape. In this paper, we study the use of related techniques to detect and classify gestures on a 3-Dimensional embodied entity, such as a robot, in the context of social touch interaction. Detecting touch on social robots poses different requirements for the geometric setup of the system, and requires a different set of gestures than user-interface devices.

Many robotic interaction systems use external facing cameras to detect and identify remote hand gestures for human-robot interaction. To name just a few, Malima et al. [24] proposed an algorithm that can recognize a limited set of hand gestures as robot control commands in real time, and was invariant to rotations and scale of the hand. Xu et al. [46] used a Kinect depth sensor to carry out 3D hand tracking which can be used to navigate a service robot. Nuzzi et al. [27] used a faster R-CNN object detector to recognize gestures of two hands with a collaborative robot setup, leading to real-time human-robot interaction with low inference time. These setups use external facing cameras on the robot, intended for detecting distal gestures. Although the proposed algorithms are related to gesture recognition, such setups are not able to directly detect touch gestures, as they can not capture the region that is in the contact with the skin. Capturing the gesture from a side view using an external camera facing parallel to the skin may provide a possible solution, but is less effective as it can not directly obtain the information such as the touch positions and contact shape from the images. Moreover, it has a limited coverage from a single perspective, for example, one hand may obstruct the camera from capturing the other hand by blocking the line-of-sight. A number of external cameras are needed to cover the full detection range of the robot's body, facing in multiple directions. In sum, an external-facing camera does not pose a practical solution to detect touches on the robot's skin.

3 SHADOWSENSE

To address the shortcomings in the existing literature of HRI touch interaction and to study vision-based methods for gesture classification and tracking in the HRI domain, we propose ShadowSense: embedding the vision sensor inside the robot and capturing shadows generated by social touch gestures of the robot's skin. The "robot" can be something as simple as a balloon (Figure 1), augmented to sense touch by placing a camera inside it.

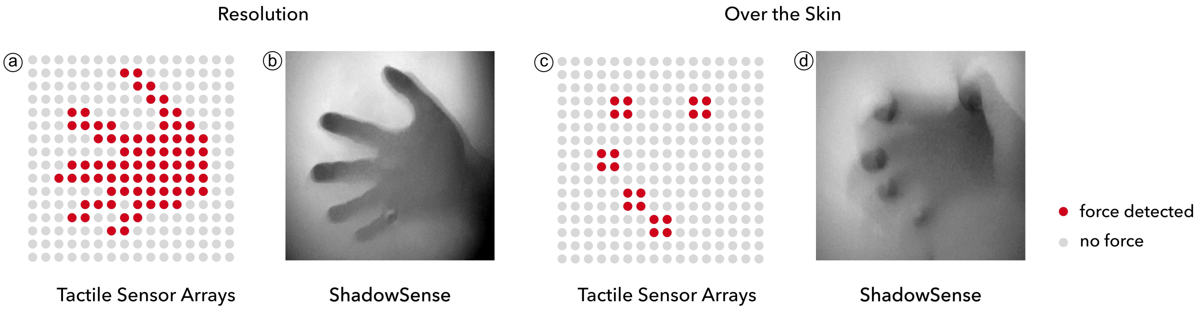

Fig. 2. A comparison between traditional sensing arrays and ShadowSense. (a) The sensing resolution of traditional methods is limited to every few millimeters, while (b) ShadowSense can achieve high resolution. (c) Sensor arrays can only detect touch with direct contact and mechanical force, while (d) ShadowSense can capture both on-skin and over-the-skin activities.

3.1 A Modality between Vision and Touch

We think of ShadowSense as a modality between vision and touch—it brings the high resolution and low-cost of vision sensing to the close-up sensory experience of touch. It outperforms the traditional sensing arrays by achieving higher resolution and gaining the ability to detect both on-skin and over-the-skin activities. Figure 2 illustrates these differences.

Force-based tactile sensors obtain information from discretely arranged sensor arrays. The spatial resolution of detection depends on how densely the sensors are arranged, usually varied from millimeter to centimeter scale. It is costly to further improve the resolution, as the number of sensors will increase quadratically with the growth of pixels in each row. ShadowSense, on the contrary, increases the sensing resolution with minimal hardware.

Traditional tactile sensors also require contact with the surface for detection. Light contact is barely detectable for many sensors such as FSRs. ShadowSense, in contrast, is not only able to detect force applied to the contact surface but expands the sensing space to include mid-air sensing, by capturing half-shadows generated from activities close to the skin. It may further differentiate between the on-skin and over-the-skin touches by analysing the blurriness and luminance of shadows, as distant shadows tends to be more blurry and project a lighter-colored image onto the robot's skin.

3.2 Interaction Capabilities

The core interaction capabilities of ShadowSense include detecting touch activity, classifying touch gestures, and identifying the position of the touch event (Figure 3).

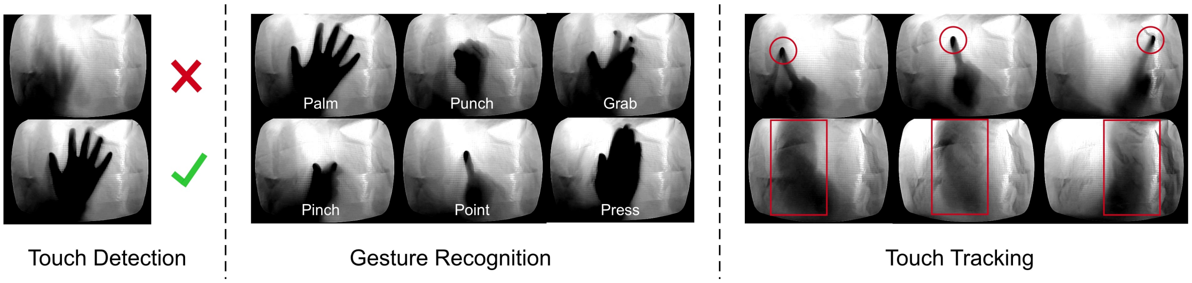

Fig. 3. ShadowSense can (a) detect touch activities, (b) classify touch gestures, and (c) track touch positions.

3.2.1 Detection. ShadowSense can be used to detect the occurrence of physical contact on its skin. It can sense when a touch operation begins and when it ends. ShadowSense is able to detect even a light touch, which applies minimal force. Beside direct physical contact, over-the-skin input can be detected as well. See, for example, the central area of the palm in Figure 2 (d) while performing a grabbing gesture. Over-the-skin detection is useful for acquiring information from non-contact body parts in a touch operation, detecting touch attempts before a physical contact actually happens, or estimating the pose or orientation of the user.

3.2.2 Classification. ShadowSense supports the classification of touch gestures when an activity is detected. This allows for analyzing the social meanings or user intentions behind a touch gesture. For example, when considering a hand, poking someone's body with a fingertip is usually meant to get their attention but punching them could be a sign of aggression. The range of gestures that can be sensed also varies in scale: from detecting a small fingertip touch, to a human-sized full body hug, depending on the scale of the skin and how the camera is arranged.

3.2.3 Shadow Tracking. ShadowSense also allows for tracking the position of shadow, further enriching the interaction possibilities. The method can sense the touch position when a touch operation begins, and keep tracking it when the touch gesture is moving. In addition, it can perform over the skin shadow tracking, for example, tracking the position of human bodies when they stand close to the skin, anticipating touch activities.

4 IMPLEMENTATION

In this section, we present an implementation of ShadowSense. We first describe a hardware prototype that captures shadows during touch interaction, followed by a description of a neural network-based algorithm that can detect and classify gestures from the shadow images, and an algorithm that tracks the position of touches.

4.1 Hardware Prototype

We built the hardware prototype in the context of a research project on robots guiding humans through a space. To provide both safety and a large-scale information display for users, the robot was designed as a human-scale soft inflatable bladder mounted on a mobile base (Figure 4).

The bladder is made of a skin of white translucent nylon fabric. It has six embedded inflatable channels, and a circular spring. The bladder is a cylindrical shape, with a diameter of 50 cm and a height of 120 cm and can hold its shape when all the six channels are inflated. An inclined plane cuts across the top of the cylinder, forming an elliptical surface that allows for touch and information display.

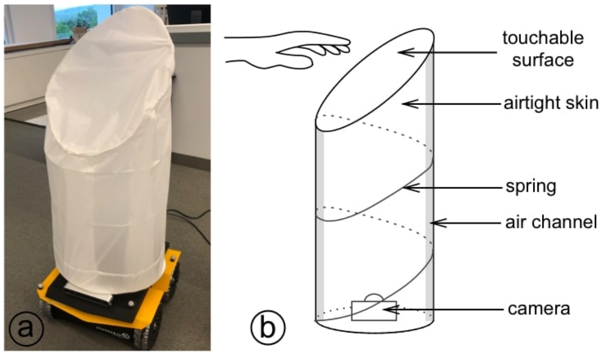

Fig. 4. A hardware prototype robot, which demonstrates the use of ShadowSense, is made of an inflatable translucent bladder on top of a mobile base.

A USB camera (*OmniVision*) with a 170 degree field-of-view fisheye lens is placed at the center of the bottom surface, facing straight upwards. The fisheye view allows it to easily capture any shadows on the top of the bladder's skin, and on parts of the side (see: Figure 6).

4.2 Detecting and Classifying Touch Gestures

Given an image of a shadow, our goal is to classify the image to its corresponding interaction gesture. Due to the variety of shadow images, we use machine learning techniques where classification models are trained with large and diverse datasets and then used during recognition [10, 29]. Convolutional Neural Networks (CNNs) [20] are widely applied in image classification. With proper transformations to the training data, they are robust against image variations, such as scale and rotation [9], making them ideal for this purpose. We use a Densely Connected Convolutional Network (*DenseNet*) architecture as proposed in [12]. This network contains shorter connections between layers, and is able to alleviate the vanishing-gradient problem and strengthen feature propagation.

In the implementation of the network, we used a pre-trained image classification model. It is based on the assumption of *transfer learning*, where a model developed for a base task is reused as the starting point for the model of a target task [39]. Research shows that transfer learning in image classification can achieve comparable accuracy to a specifically trained model, while saving time, data and resources [30]. The idea is that the earlier layers of the network are trained on a massive corpus of photographs, and that only the final layers are then re-trained on the target dataset. As low-level features tend to be common to many datasets, the process will work for both base and target classification tasks. This approach is particularly useful in our case, as there is no publicly available dataset of shadow touch images. Given the custom construction of our robot, we would have to collect interaction images with this specific robot geometry, which would be prohibitively expensive.

In our prototype, we used the pre-trained model *DenseNet-161* [12] because of its high accuracy in classifying images from a large variety of categories, and its low number of parameters which allows for fast prediction. Figure 5 presents the structure of the network.

The structure of the DenseNet-161 used is as follows: A convolution layer is followed by four dense blocks with transition layers, followed by a fully connected classifier. The four dense blocks consist of 6, 12, 36, and 24 dense layers in order from input to output. Each dense layer has two batch normalization layers, two ReLU activations, and two 2D convolutional layers. Transition layers connect two adjacent dense blocks, changing feature map size via convolution and pooling.

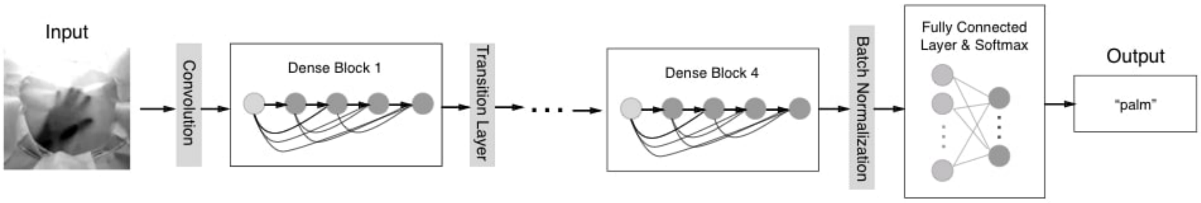

Fig. 5. The network we used, based on DenseNet-161 [12], takes a shadow image as an input and predicts a gesture label.

A fully-connected classifier with a softmax output unit computes a probability distribution over the class labels as the last stage of the network. To adapt the model to fit our task of interest, we train the weights of this last-stage classifier by training it using labeled shadow images. Section 5.1 presents an experiment that trains and tests the network with shadow images of six gestures.

4.3 Tracking Touch Positions

In addition to classifying the gesture, we also want to estimate the position of a touch activity when it is detected. Our approach is inspired by the work of using color segmentation to perform skin detection. Yang et al. [47] present a skin-color segmentation algorithm that characterizes the skin color of human faces, and is able to track a human face in real-time in various poses and views and can be adapted for different people and lighting conditions. Using color parameters calibrated ahead of time, we perform touch tracking from the images. As contact touch usually leaves a stronger shadow than a hovering hand, we separately segment shadow and contact skin by applying two different color thresholding masks. This allows to differentiate between the touch gesture that is in direct contact and that above the skin. We refer to these two separate detection as "shadow" and "contact" below. The sets of RGB ranges used for the shadow and contact color classes were determined from images that were not included in the evaluation of the algorithm.

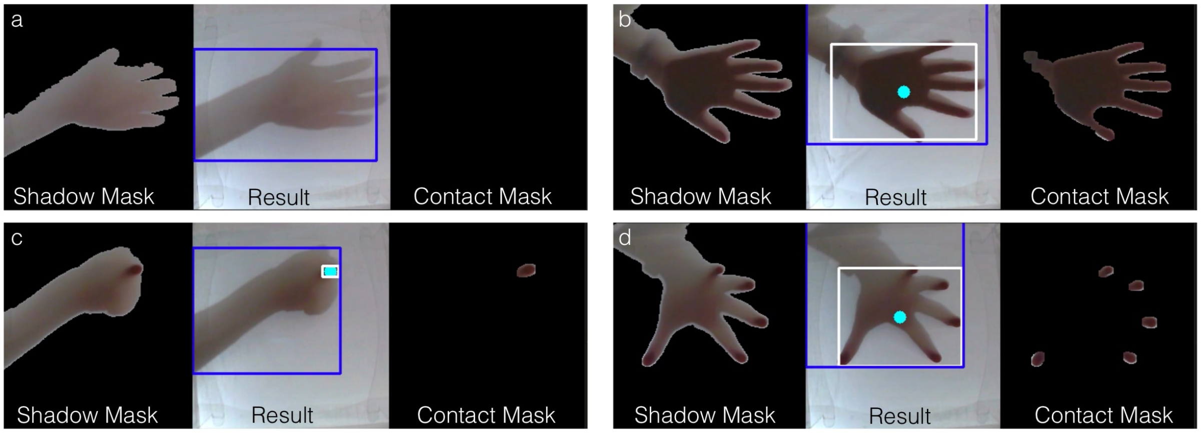

Fig. 6. The tracking results on a (a) hover gesture, (b) palm touch, (c) fingertip touch, and (d) grab. Images on the left depict the contours after segmenting the color of shadow, and images on the right are contours after filtering the color of contact skin. Bounding boxes in the middle draw the tracking results of contact areas (in white), and overall shadows (in blue).

Filtering the RGB values of pixels by matching the color of the shadow or contact results in a binary image representing shadow or contact locations. A Gaussian blur function is used to smooth the binary image resulted from color filtering. Contours of the image are found using the Suzuki and Abe topological contour following algorithm [37]. We use the `findContours` implementation provided by the `openCV` library. Contours are thresholded by size and nearby contours are combined to form the shadow and contact areas. Figure 6 shows the contours after applying the shadow mask (left) and the contact mask (right), for four touch gestures: (a) hover, (b) palm touch, (c) fingertip touch, and (d) grab.

After contour detection, we fit a bounding box around the contours (Figure 6 center), separately for the shadow area (blue box) and contact area (white box). The touch position is estimated based on the centroid of the contact region. Separately tracking the overall shadow and the direct contact locations allows the robot to identify and track a wide range of touch gestures, from those that are in direct contact with the skin, to touch attempts that hover above the skin, and non-contact body parts during a touch operation, for example, the arm, and the non-contact hand areas during a point or a grab gesture in Figure 6 (c) or (d).

We further process the outputs of the above tracking algorithm into a finite state machine of touch states, in order to infer interaction context. The system transits between five states: *none, appear, touch, move*, and *lift*, activated by events indicating the detection, movement and loss of shadow or contact positions. Figure 7 visualizes the states and the events that activate the state transitions. The system starts from a default state of *none*; when a shadow is detected, it transits to an *appear* state and which keeps track of the shadow position. The detection of contact position transitions to the *touch* state. We added a 150ms hysteresis window in order to reduce false transitions.

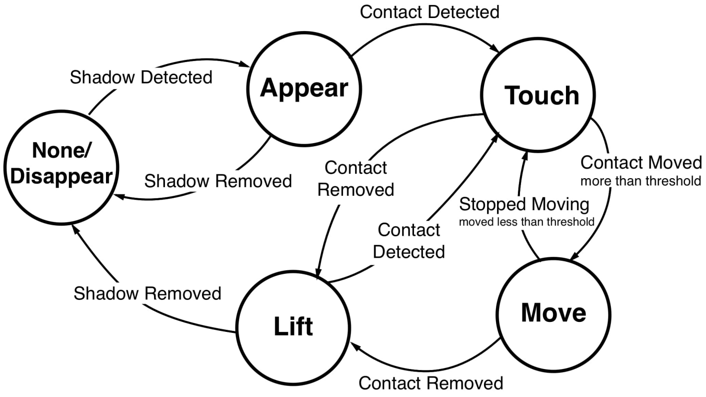

Fig. 7. The state machine visualizes the five states (circles), and the events (arrows) that activate the state transitions.

Contact positions are differentiated and the differential is compared to a fixed threshold to transition from a *touch* towards a *move* state. The direction of the motion is calculated by averaging the displacement on a time window of 300ms, and categorized into one of four directions ("Left", "Right", "Front", and "Back") in the robot's coordinate frame. The state changes back to *touch* if the position differential is below the movement threshold. The removal of direct contact transits to a *lift* state, whereas the removal of the overall shadow further changes to the *none* state. We also apply a 150ms hysteresis time window before changing to a *lift* or a *disappear* state. Figure 8 presents a temporal sequence of frames capturing the full action of a finger appearing and sliding along the surface. The state changes in the following order: *none*, *appear*, *touch*, *move*, and *lift*, activated by the change

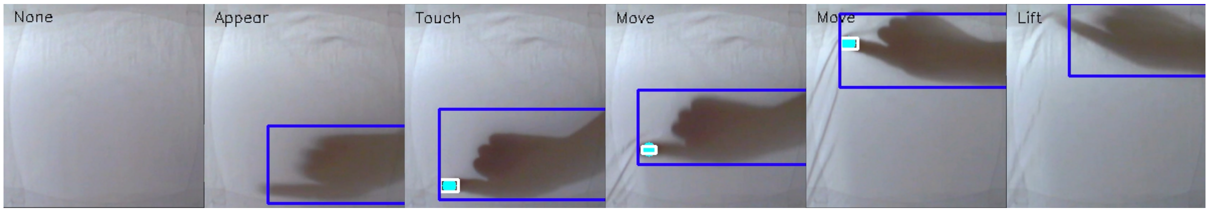

Fig. 8. A temporal sequence of frames capturing the action of a finger sliding along the surface. The tracking results are represented with the bounding boxes, and are used to activate state transitions. The state of each frame is displayed on its top left corner.

of shadow and touch positions, indicated by the bounding boxes and centroid circle. Section 5.2 quantitatively evaluates the tracking performance on image frames and video sequences.

5 TECHNICAL EVALUATION

We quantitatively evaluated the performance of the gesture classification and the tracking algorithm in a series of validation studies, with the aim of assessing their accuracy in contexts similar to those encountered in HRI social touch scenarios.

5.1 Evaluation of the Gesture Classification Algorithm

We first present an evaluation of the performance of the ShadowSense recognition method. Social touch images were collected using our hardware prototype and were used to train the DenseNet described in Section 4.2. We present results from classifying test images collected under three lighting conditions.

5.1.1 Data Collection. The dataset used in training the network is made up of image frames of shadows captured from the fisheye camera described in Section 4. To collect the images, the authors, along with other volunteers in the lab, performed the following actions on the upper body of the robotic bladder, illustrated by the top row of Figure 9: staying close but not touching any part of the robot (a), hugging the robot (b), touching the surface with a palm (c), pointing on the surface with a fingertip (d), punching the surface with a fist (e), and touching the surface with two palms (f). We recorded the different gestures to create labeled shadow images.

To collect a variety of interaction data, we had four different people (two male, two female) of varying heights (160 -178cm) and hand lengths (15.7 - 19.1cm) performing the same gestures. To test the system under diverse environment conditions, the data was collected under three lighting conditions: daylight, dusk, and night.

We collected a total of 6,120 RGB images of size 635 × 350 pixels, divided equally in 6 gesture categories. Figure 9 shows samples from the training dataset.

5.1.2 Data Preprocessing. Each shadow image was resized to 318 × 235 pixels in order to reduce the training time. It was then cropped with a window of 220 × 150 pixels, leaving only the central area that captures the contact shadow. To allow the shadow to be clearly seen, we adjusted the contrast of the image by a factor of 3 using the *torchvision.transforms* module in the PyTorch machine learning framework. For the images collected in a dark environment, we adjust the brightness by a factor of 2 to lighten the images. Figure 10 displays two examples of the raw image data and images after adjustment. Finally, we normalized the image with $mean = [0.485, 0.456, 0.406]$, $std = [0.229, 0.224, 0.225]$ before using them in the classifier network. These values are determined by the design of the pre-trained early-stage classification network.

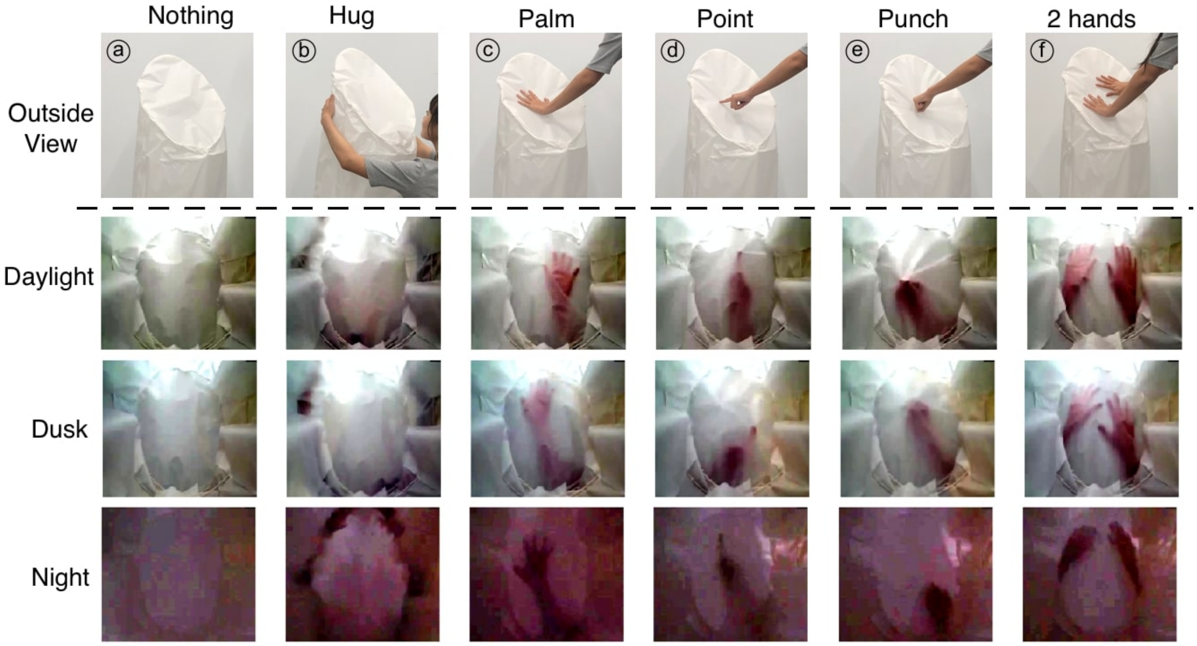

Fig. 9. The six interaction gestures toward an inflatable bladder that were evaluated in our experiments. The top row captured interaction gestures from an external view, whereas the bottom three rows are the image samples from training dataset, captured by an internal camera under three lighting conditions. Images are presented after preprocessing.

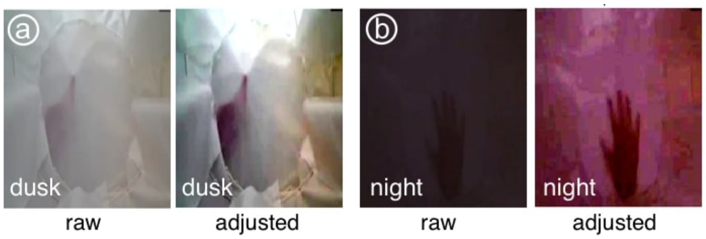

Fig. 10. To preprocess the data, we adjusted (a) the contrast and (b) the brightness of the images.

5.1.3 Network Implementation. We used the open-source PyTorch machine learning framework to implement the neural network and trained it on a cloud-based GPU (*Google Colab*). We used the *Adam* [19] optimizer with learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-8}$, weight decay of 0, and negative log-likelihood loss as the objective. A 5-fold cross-validation was applied to tune the number of training epochs, resulting in 20 epochs with a batch size of 16.

5.1.4 Results on a Combined Dataset. We trained the network with a combined dataset, covering all the three lighting conditions and the four users. The dataset of each gesture and condition was split by the order of collection time to allocate 70% of the data to training and 30% to testing, leading to a total of 4284 images in the training set and 1836 images in the test set.

Table 1. Classification accuracy under three lighting conditions.

Lighting Conditions	Daylight	Dusk	Night	Average
Accuracy	96.04%	92.94%	87.46%	92.15%

The average classification accuracy under each lighting condition was calculated separately, and is presented in Table 1. Shadow images captured in daylight were classified with the highest accuracy (96.04%), followed by dusk (92.94%). Night gestures were classified with the lowest accuracy (87.46%).

Figure 11 shows a confusion matrix was obtained for each lighting condition, illustrating the classification accuracy (%) for each gesture. The diagonal-distributed pattern in the confusion matrix shows the classifier is able to distinguish between the six touch gestures under all lighting conditions with a relatively high accuracy.

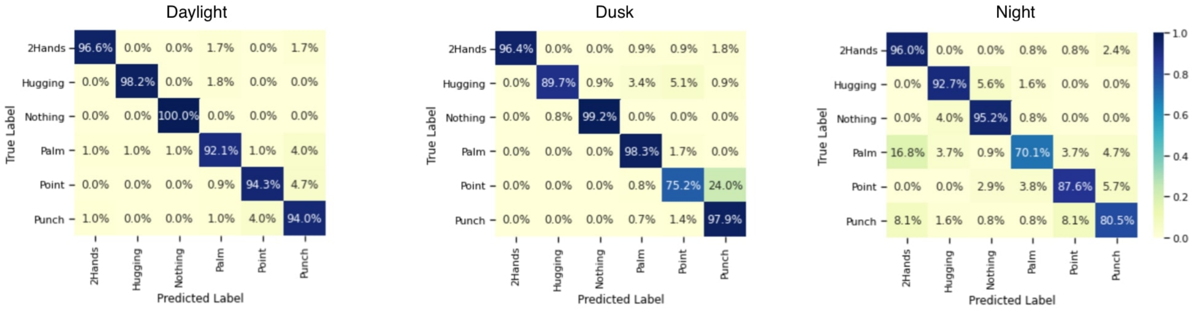

Fig. 11. Confusion matrices for classification (in percentage), by lighting condition.

Under night lighting, "palm" has the lowest recognition rate, and is easily confused with "two hands". Furthermore, the gesture "point" is the most difficult to classify under dusk, and the third most difficult under daylight and night, where it is mostly confused with "punch". Finally, "hugging" and "nothing" are slightly confused with each other at night, as it may become harder for the system to capture the arm and hand in the dark lighting conditions.

5.1.5 Generalization to Unseen Users. We further trained and tested the classification algorithm with holdout sets of unseen users. This allows to test the robustness of the algorithm and assess its ability to generalize to users not present in the dataset.

We iteratively sequestered one of the four participants' data as a hold-out test set. Then we trained a model for 20 epochs using the remaining three participants' data, with a 70:30 training/validation split. The epoch with highest validation accuracy was selected as final model to evaluate the hold-out set. This procedure was repeated with each participant used once as the hold-out participant. The classification accuracy on each hold-out user and their average are presented in Table 2. The algorithm achieves a comparable accuracy on test data of a new user (87.90%), but is slightly lower than when the model was trained and tested on the full data set.

Table 2. Test accuracy on a hold-out user

Hold-out	User 1	User 2	User 3	User 4	Average
Accuracy	86.41%	88.23%	85.83%	87.86%	87.90%

Table 3. Test accuracy on a hold-out lighting condition

Hold-out	Daylight	Dusk	Night	Average
Accuracy	67.35%	63.19%	30.83%	53.79%

5.1.6 Generalization to Unseen Lighting Conditions. We also evaluated the algorithm by iteratively holding out data collected under one of the lighting conditions, daylight, dusk, and night. We used the remaining data to train a classifier following the method above, and tested on the hold-out data. Table 3 presents classification accuracy on each hold-out test set and their average. The method did not obtain similarly generalizing performance on images from unseen lighting condition (best: 67.35%, mean: 53.79%), especially when the hold-out set was taking at night.

In order to gain some insights of the visual differences between the gestures under different lighting conditions, we present the confusion matrices on each lighting in Figure 12. Some gestures were classified fairly well when holding out daylight or dusk data, for example, the gesture "hugging" and "two hands". Gestures operated with a single hand but different hand configurations, such as "punch", "point", and "palm", are harder to be differentiated if has not been seen under this specific lighting. When tested using the holdout data of night, the classifier biased the classification results heavily toward some gestures: "punch", "two hands", and "nothing", whereas no gesture has been classified as "hugging" or "point".

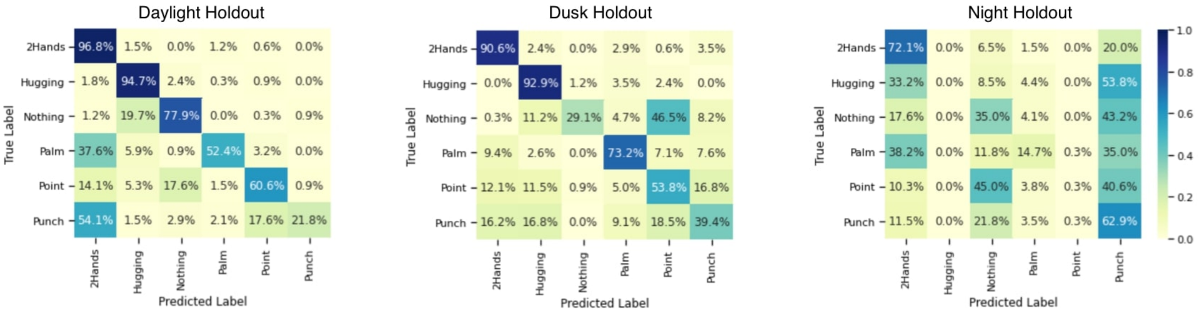

Fig. 12. Confusion matrices for classification (in percentage), each evaluated on a hold-out lighting condition.

To further understand the ability to generalize learned models between lighting conditions, we iteratively trained a classifier using the data collected under one lighting condition, and tested with the data from a different lighting. The pairwise results are presented in Table 4. Each column represents the lighting condition that was used for training the classifier, and each row represents the lighting condition that was used for testing the classification accuracy. The results further indicate that models trained on one lighting condition do not easily generalize to a different lighting context, especially when one of the data sets is taken at night. Still, the learned model generalized better between daylight and dusk.

5.2 Evaluation of the Tracking Algorithm

This section describes experiments to evaluate the performance of the tracking algorithm. We first evaluate the accuracy of tracking shadow and contact positions on image frames, and then test the accuracy of inferring motion activity from a video.

Table 4. Classification accuracy with the classifier trained on one lighting condition and tested on another

Classification Accuracy		Train		
		Daylight	Dusk	Night
Test	Daylight	N/A	80.88%	14.66%
	Dusk	54.70%	N/A	25.50%
	Night	19.17%	27.21%	N/A

5.2.1 Data Collection. The experiments were conducted using a transparent plastic box of 26cm × 26.6cm width and length, and 45cm height, covered with white translucent nylon fabric. A fish-eye camera was placed at the center of the bottom surface, facing upwards.

A researcher performed five categories of gestures on the platform, including (a) *nothing*: standing in front of the platform at least one meter away from the surface, (b) *hover*: hovering a hand or an arm above the surface, at a distance between 5 to 20 cm, (c) *palm*: touching the surface with a down-facing palm, (d) *fist*: touching the surface with a fist, and (e) *finger*: touching the surface with a fingertip. The above gestures were performed using a single hand. To increase the diversity of test images, we performed gestures with a variety of touch or hover positions, with different hand orientations, and positioned the test platform in different environments. A total of 500 RGB image frames were collected, equally divided to the five categories. The image frames were resized and cropped to the size of 200 × 205 pixels before feeding into the tracking algorithm.

5.2.2 Tracking Performance Test on Image Frames. Using the method presented in Section 4.3, we separately track the positions of shadow and contact areas. Figure 13 presents the tracking results on example images in the five categories. We use bounding boxes to represent tracking results of shadow areas (first row, blue color) and contact areas (second row, white color). To quantitatively evaluate the performance, we manually labeled the 500 images with ground truth bounding boxes (shown in red). The ground truth bounding boxes were determined ahead of time, before evaluating the performance of the algorithm, using the following rules: Shadows were defined using the minimum enclosing rectangle of all the visible shadows created by human body parts, including hand, wrist, and the arm. Contact areas were defined as the minimum enclosing rectangle of areas that could be visually identified as having direct contact with the surface. We then compared the tracking results with the ground truth to measure the tracking accuracy.

We use three metrics to measure the accuracy of the tracking algorithm. As in [25], we first measure the error in the position (x, y) of the center of the bounding box, compared to the ground truth. We then measure how closely the two bounding boxes match using the "Intersection over Union" (IoU) measure. This measure calculates the overlapping area of intersection between two bounding boxes, divided by the total area of both bounding boxes. A higher IoU score represents a better match, with 1.0 representing a perfect match. We finally report the error rate of the tracking process, that is, the percentage of frames when an existing shadow or contact have not been detected.

Table 5 presents the statistical accuracy of the tracking algorithm, separate for shadow and contact tracking of the five gesture categories. Besides the results presented in the table, there was no false positive detection of shadow or contact in gesture *nothing*, nor false detection of contact in gesture *hover*. Overall, the tracking algorithm performs well for the gestures we evaluated. When tracking the positions of shadows, gestures in closer distance with the surface are more precisely tracked. The tracking of the *hover* gesture results in higher error in detecting its position, and 6% of *hover* shadows were undetected. The overall shadows of all the closer-to-surface gestures (*palm, fist, finger*) were successfully tracked, with less than 0.2 centimeter error in tracking its center position, and above 0.93 IoU ratio. However, the tracking of contact positions of those gestures were slightly less

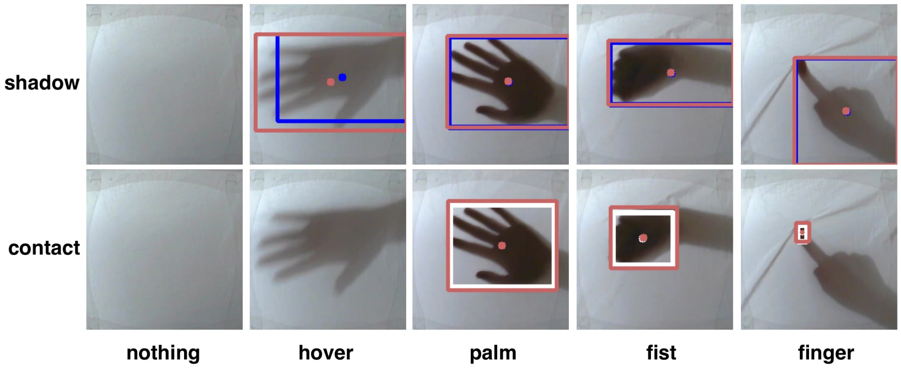

Fig. 13. Results of tracking shadow (1st row) and contact (2nd row) positions on five image frames, each for one gesture. The tracking results are represented using bounding boxes in different color (blue for shadow, and white for contact). The ground truth bounding boxes are constructed manually for each frame (in red color).

Table 5. Statistical accuracy of the proposed tracking algorithm in terms of the mean error, *IoU* score, and standard deviation (in brackets) of each tracking category.

Mean (SD)	shadow tracking				contact tracking		
	hover	palm	fist	finger	palm	fist	finger
error in X direction, cm	-0.41 (1.78)	-0.15 (0.24)	-0.09 (0.07)	-0.12 (0.07)	-0.19 (0.74)	-0.41 (0.50)	0.03 (0.19)
error in Y direction, cm	-0.55 (2.01)	-0.16 (0.19)	0.02 (0.15)	-0.16 (0.15)	-0.27 (0.49)	0.09 (0.44)	0.01 (0.12)
Intersection over Union	0.66 (0.20)	0.95 (0.03)	0.96 (0.02)	0.93 (0.04)	0.82 (0.11)	0.76 (0.12)	0.51 (0.17)
tracking error (%)	6%	0%	0%	0%	0%	0%	14%

accurate compared to the tracking of their shadow, especially for the gesture *finger*, which has the lowest *IoU* ratio and is more likely to be undetected (14%).

5.2.3 Evaluation of Motion Interpretation on Video Sequences. We also evaluate the performance of inferring motion activity by tracking a sequence of image frames. To contextualize the evaluation in a real human-robot interaction application, we imagined an intuitive interaction scenario, where a user slides along the robot's surface to instruct the robot to move towards a corresponding direction. To make the evaluation more feasible, we assume a discrete set of motion directions the robot can take (*moving forward, backward, towards left,towards right*), and the user can only slide along one of the four defined directions in each motion sequence.

We demonstrate the results of our proposed algorithm on a set of sliding motion clips with a frame rate of approximately 13 fps at 200×205 pixels. A total of 160 video samples were collected, with a performer conducting sliding gestures along the four directions, with either a palm or a fingertip. Each video sample started with capturing a hand making contact with the surface at a random position, and sliding towards a direction for around 10-26 cm distance. The performer conducted the sliding gestures of varied sliding speed, with the length of the video varying between 0.6 and 6 seconds (8 - 80 frames). The 160 video clips had 40 for each of the four directions, and were made up of two sliding gestures (palm or finger), with 20 videos in each condition.

Taking a video as an input, the tracking algorithm presented in Section 4.3 is used to detect touch states and track contact positions. We set the motion threshold to 2 pixels before transiting to a *move* state. Once in a *move* state, a sliding window of 300ms (4 frames) is used to calculate the average displacements along x and y axis, and infer the motion direction by taking the direction with the highest displacement. A basic voting approach is then used on the results of all the time windows to infer the motion direction of a video. The result is then compared to the manually labeled ground truth to evaluate its performance.

In order to test the system's ability to identify motion and its direction at their early stage, we measured the system's performance for detecting motion direction of incomplete sliding gesture executions. The experiment was conducted with 10 different observation ratio setting, from 0.1 to 1.0, representing the fraction of the frames from the beginning of the clip until inference.

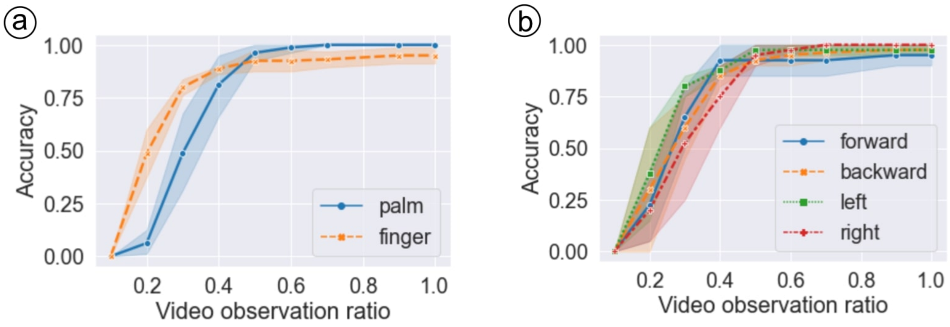

Fig. 14. Motion recognition performance with respect to the observed ratio of tested video, averaged by gestures(a) and direction of motion(b).

Figure 14 illustrates the performance curves of the implemented system. The x axis corresponds to the observed ratio of the testing videos, and the y axis to the accuracy on recognizing the motion activity and its corresponding direction from the observed video. The graphs show averaged performance for each sliding gesture (Figure 14 (a)) and sliding direction (Figure 14 (b)), over all 160 video clips.

The figure confirms that the proposed method is able to recognize the motion and its direction at a relatively early stage of the video. The method is able to make an inference with the accuracy of above 0.9 after observing the first 50% of the video sequence. In addition, palm gestures have a higher recognition accuracy than finger gestures (100% vs. 95%), after observing the full length of the video (observation ratio 1.0). The system failed to identify the contact of the finger in four (4) videos (5%). Still, the sliding gesture performed with a finger achieves better performance at earlier stage (the initial 50%) than a palm gesture. There is no systematic difference of the recognition accuracy between sliding directions.

6 POTENTIAL HRI APPLICATIONS

Our results show that ShadowSense is a promising method for implementing full-body social touch for human-robot interaction. It can achieve high recognition, tracking, and movement inference rates in realistic conditions, tested on an inflatable robot prototype and on a stand-alone testing device. ShadowSense thus promises to be a way to easily add the possibility of social touch to the interaction space of any robots or objects with an inflatable or translucent skin.

This can also lower the barrier for developing touch-enabled social robots or minimalist robotic devices. For example, Figure 15 shows a concept sketch of adding ShadowSense to a balloon. Within minutes, the balloon

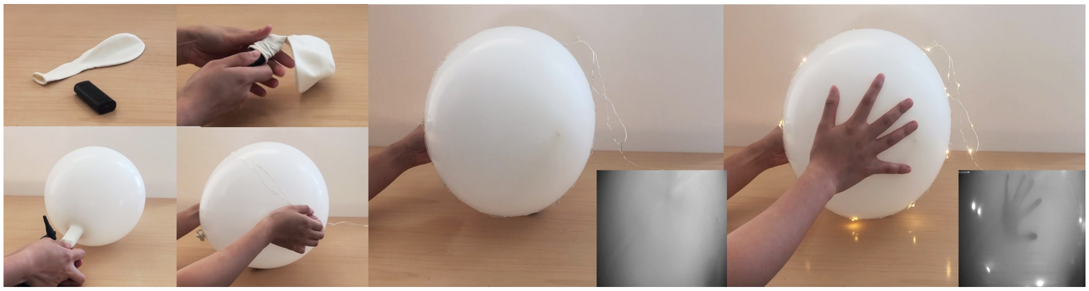

Fig. 15. Concept sketch of adding ShadowSense to a balloon. The balloon could become a touch-sensitive object, responding to touch by illuminating an LED strip.

could become a touch-sensitive object, which can respond by illuminating an LED strip connected to an off-board microcontroller.

6.1 Touch Gestures for Mobile Robot Guidance

In our own design of the inflatable mobile guide robot, shown in Figure 16, we explored the following interaction concepts using ShadowSense. Using the classification and tracking approach described above, the robot platform could respond to various touch gestures and make movements accordingly. For example, when the robot detects a “poke”, it could turn around to face the human (shown in subfigure (a)); a ”sliding” gesture could instruct it to move in the corresponding direction (b); and a tap on the back would send it on its way (c).

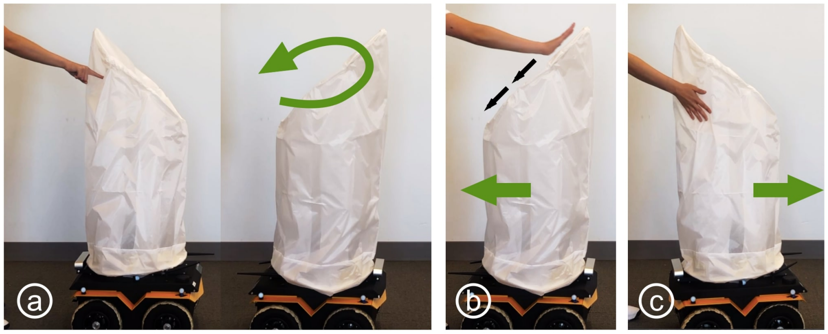

Fig. 16. Concept sketches for interaction using ShadowSense on an inflatable mobile robot. (a) When the robot detects a "poke", it turns around to face the human. (b) A "sliding" gesture instructs the robot to move closer. (c) A tap on the back sends it on its way.

Beyond social touch gestures, we propose two additional application areas in the HRI domain that could make use of ShadowSense: interactive touch screens on inflatable robots, and privacy maintaining home assistant robots.

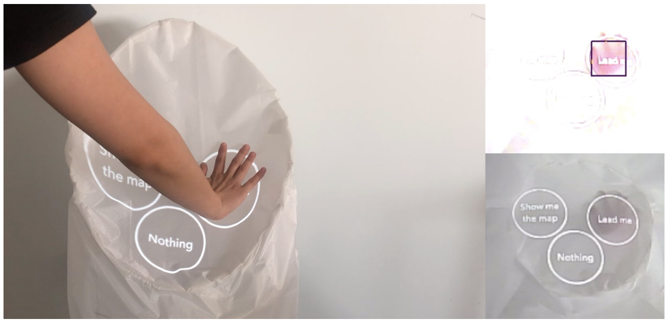

Fig. 17. A robot's skin can be augmented into an interactive touch screen by integrating ShadowSense with a projector. Users can instruct robot by pressing the virtual buttons.

6.2 Interactive Screens on Soft Robots

Using ShadowSense in combination with an internal projector, any inflatable robot can be augmented to include an interactive touch screen. Figure 17 shows this application use-case. An upward facing projector is fixed inside the translucent bladder at the center of the bottom surface. The projector casts "hot zones" onto the robot's skin, with each zone connected to one of the robot's behaviors. Users can select any of the choices by pressing this virtual button. We have implemented such an interface using ShadowSense in combination with the method described in Section 4.3.

6.3 Privacy for Home Assistant Robots

An additional use-case for ShadowSense is to enable interaction with a home assistant while avoiding the privacy-invading presence of a camera. Many social robots use a camera as an input to monitor a user's state or intentions. However, when entering a personal space like a home, having a camera can pose risks to the user's privacy. ShadowSense could provide a privacy-maintaining alternative to camera-based interaction with social home robots. By physically covering the robot's eyes with a translucent material, as some users already do with their laptop cameras, a robot can still make use of some interaction data in the form of user's shadows instead of high-fidelity images.

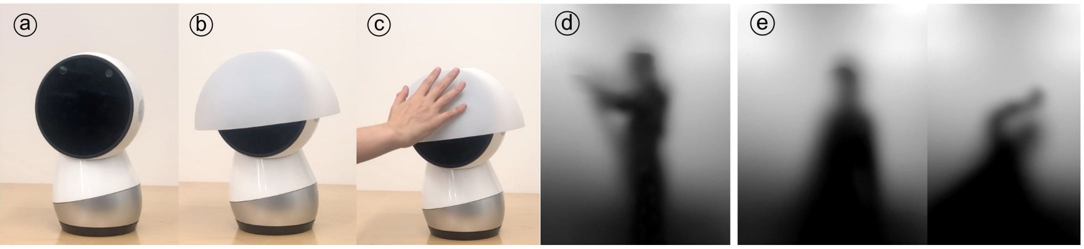

Fig. 18. ShadowSense equips a camera-based social robot (a) with the option for a privacy shield. The robot's head is covered with a translucent shell (b–c), shifting its input channel from visual to tactile. Being covered with translucent material (d–e), the robot may infer the users' activities without seeing their full appearance.

Figure 18 shows concept sketches demonstrating the idea of a privacy shield. A home robot in Figure 18 (a) normally has a camera capturing images inside the home. A user can cover its camera with a translucent shield (b) but can still interact with the robot using ShadowSense touch (c). The robot's camera is now a touch sensor, allowing users to engage with the assistive robot on their own terms.

Furthermore, while the robot is prevented from capturing a clear image of users or the surrounding environments, it can still monitor some non-tactile aspect of users' activities. Figure 18 (e) - (f) shows the robot's view through a privacy-screened camera: it is possible to infer what a user is doing from shadows without collecting high-resolution images of their appearance. Such a robot could detect important events such as accidental falls. Users can balance the trade-offs between the effectiveness of privacy preserving and the ability to detect the activity information by adjusting the material of the shield. Low translucency materials prioritize the privacy aspect but result in limited detection range and the capability to only classify simple activities. Higher translucency could be more accurate, but might encroach more on the user's privacy.

Giving users a choice to choose a physical cover in front of robot's eyes may help them feel safer and more comfortable with living and interacting with a robot in a personal space, without having to accept an all-or-nothing trade-off between privacy and functionality. It is left for future work to develop and evaluate this application in order to make improvements to the shadow detection and recognition algorithms. This would include methods for the processing of full body shadow images and the interpretation of users' actions, as well as user studies evaluating the benefits of privacy-preserving interactions.

7 DISCUSSION AND LIMITATIONS

ShadowSense is a vision-based tactile sensing method that operates by capturing contact shadows from a view inside the robot. We see this method as a sensory modality that lies in-between vision and touch. Just as humans are able to detect some light and motion information with their eyes closed, we propose a method that is akin to "eyes" under the skin: instead of being sensitive to pressure, ShadowSense receptors are sensitive to light changes. Metaphorically, the eye-under-the-skin replaces millions of pressure receptors in the skin, as each pixel becomes a remote touch receptor on the skin's surface.

In practice, shadow-based touch sensing has several characteristics that enable it to outperform the traditional approach of pressure or contact sensor arrays. First, it is not as constrained by skin geometry and can operate with curved, soft, or deformable exterior surfaces. Second, the information acquired is of a much higher resolution than the traditional sensor arrays, detecting millions of pixels of sensor readings on a single skin surface. Finally, more types of touch are detectable, including light touches, close-to-skin touches, touches where only part of the hand is in contact with the robot's skin, and even touch attempts that happen before physical contact occurs.

We argue that using a shadow-based vision method on tactile sensation will pave the way to a new generation of tactile sensors in robotics, especially in soft robotics. Still, some limitations remain with our proposed approach.

7.1 Limitations Inherent to Computer Vision

Replacing force or contact-based skin receptors with visual sensors carries some intrinsic drawbacks. Some of these drawbacks are related to the visual modality and are common to all computer vision methods. Lighting conditions and environmental noise can cause fluctuations in the sensing capabilities. In this paper, we found that darker lighting conditions can negatively effect the classification accuracy of gestures. Environmental noise, such as shadows created from the surrounding objects, or wrinkles on the skin surface can also interfere with the classification. In addition, putting the camera too close to the skin may restrict the sensing area.

The proposed approach requires leaving the space between the camera and sensory surface clear. In fact, our method assumes a nearly empty internal structure of the robot, affording a clean line-of-sight between the camera and the robot's external skin. In reality, other internal mechanisms may interfere by blocking the line-of-sight.

One possible approach to address this is to embed internal convex mirrors or other reflective materials to enlarge the field of view and avoid obstruction.

ShadowSense also requires designers to choose an appropriate field-of-view of the camera with respect to the requirements of the sensing scale and resolutions. Additional lenses, such as a fish-eye lens can be used to enlarge the view angle.

7.2 Missing Aspects of Touch

Although we can get a large portion of contact information through the shadow images, some tactile aspects are missing. For example thermal receptors on the robot's skin can detect the temperature of the contact object, pressure receptors can get a more precise estimate of the force applied to the skin. Such information is not available using ShadowSense. While ShadowSense provides some touch sensing in a simple, low cost way, if more accurate information such as temperature and fine pressure is needed, other types of sensors have to be added to the system.

In addition, ShadowSense has a single point of failure, whereas multi-sensor arrays could continue to function if only one of the sensors malfunctions.

7.3 Limitation of Data Collection for Evaluation

The evaluation described in Section 5.1 concluded that the classifier learned under one lighting condition did not generalize to a different lighting condition. This could be because the lighting setups we chose to collect data of were limited and sparsely distributed. The lighting conditions we used may not be close enough to let them share similar color distributions of each gesture nor generalize the learned features across each other. We believe that by collecting more data from a large variety of lighting conditions may help the algorithm to more precisely extract the underlying features of each gesture, invariant to external lighting, and has the potential to better generalize to unseen contexts. It is also possible to address the problem by choosing a better preprocessing method that reduces the illumination difference and normalize the color distributions across different lighting conditions or train separate models for each lighting condition and then use these models only for the conditions for which they were trained. We leave this improvement for future work. Moreover, the number of participants included in the evaluation is limited due to the constraints of in-person experiments posed by the pandemic. Future work will include testing the robustness of our method by evaluating the algorithm with a larger number of participants.

7.4 Better Classification through Motion Data

In this paper, we used Densely Connected Convolutional Neural Networks to classify touch gestures from a single shadow image frame. However, motion information could provide for more accurate gesture recognition, for example, differentiating between ”scrub” and ”pat” which may be otherwise both classified as a palm touch using the single-frame classification method. In future work, we plan to use recurrent neural networks for sequential image data, to uncover dynamic touch gesture information.

In the current implementation, we use two separate algorithms for gesture classification and touch tracking. Taking into account temporal information in gesture classification could allow us to combine the two above-mentioned functions into a unified algorithm, i.e. an object detection algorithm that could simultaneously classify as well as localize the touch operation, and infer motion parameters. This is also left for future work.

8 CONCLUSION

We presented ShadowSense, a technique to enable touch sensing for social robots using a vision-based method. The main contribution of the paper is the idea of capturing contact shadows from a view inside the robot for touch recognition, thus providing a low-cost, hardware-light alternative to traditional touch-sensing technologies applied to social robots. This technology also provides rich information when compared to existing force or capacitive sensor arrays: it is able to sense light touches as well as close-to-skin activities, achieves high-resolution full-body sensing, and applies to robots with different size and shape of surfaces. While vision-based methods have been used in 2D user-interface contexts to replace pointers operating bitmaps, we are not aware of any applicable solution for the HRI social touch contexts described above. Moreover, ShadowSense points to a new direction in tactile sensing for soft and inflatable robot design, where the surface-embedded sensors are hard to use because of skin deformation during robot activities, and the interference with the original tactile feeling of the surface material.

In this paper, we have demonstrated high recognition, tracking and movement inference rates of ShadowSense technology in realistic interaction conditions. We believe that this method could pave the way for cost-effective and feasible human-robot interaction, and enable social robots to take full advantage of the rich communicative modality of touch.

ACKNOWLEDGMENTS

This research was supported under NSF National Robotic Initiative Award NRI:1830471.

REFERENCES

[1] Alessandro Albini, Simone Denei, and Giorgio Cannata. Human hand recognition from robotic skin measurements in human-robot physical interactions. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4348–4353. IEEE, 2017.

[2] Fernando Alonso-Martín, Juan Gamboa-Montero, José Castillo, Álvaro Castro-González, and Miguel Salichs. Detecting and classifying human touches in a social robot through acoustic sensing and machine learning. *Sensors*, 17(5):1138, 2017.

[3] Alexander Alspach, Joohyung Kim, and Katsu Yamane. Design and fabrication of a soft robotic hand and arm system. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 369–375. IEEE, 2018.

[4] Kerem Altun and Karon E MacLean. Recognizing affect in human touch of a robot. *Pattern Recognition Letters*, 66:31–40, 2015.

[5] Patrick Chiu, Chelhwon Kim, and Hideto Oda. Recognizing gestures on projected button widgets with an rgb-d camera using a cnn. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, pages 369–374. ACM, 2018.

[6] Pak-Kiu Chung, Bing Fang, and Francis Quek. Mirrortrack-a vision based multi-touch system for glossy display surfaces. *IET*, pages 571–576, 2008.

[7] KC Dohse, Thomas Dohse, Jeremiah D Still, and Derrick J Parkhurst. Enhancing multi-user interaction with multi-touch tabletop displays using hand tracking. In *First International Conference on Advances in Computer-Human Interaction*, pages 297–302. IEEE, 2008.

[8] Siyuan Dong, Wenzhen Yuan, and Edward H Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 137–144. IEEE, 2017.

[9] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 643–650. ACM, 2015.

[10] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[11] Matthew J Hertenstein, Julie M Verkamp, Alyssa M Kerestes, and Rachel M Holmes. The communicative functions of touch in humans, nonhuman primates, and rats: a review and synthesis of the empirical research. *Genetic, social, and general psychology monographs*, 132 (1):5–94, 2006.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[13] Isabella Huang, Jingjun Liu, and Ruzena Bajcsy. A depth camera-based soft fingertip device for contact region estimation and perception-action coupling. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8443–8449. IEEE, 2019.

[14] Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki Kanda, and Ryohei Nakatsu. Robovie: an interactive humanoid robot. *Industrial robot: An international journal*, 28(6):498–504, 2001.

[15] Sooyeon Jeong, Kristopher Dos Santos, Suzanne Graca, Brianna O'Connell, Laurel Anderson, Nicole Stenquist, Katie Fitzpatrick, Honey Goodenough, Deirdre Logan, Peter Weinstock, et al. Designing a socially assistive robot for pediatric care. In *Proceedings of the 14th international conference on interaction design and children*, pages 387–390. ACM, 2015.

[16] Micah K Johnson and Edward H Adelson. Retrographic sensing for the measurement of surface texture and shape. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1070–1077. IEEE, 2009.

[17] Micah K Johnson, Forrester Cole, Alvin Raj, and Edward H Adelson. Microgeometry capture using an elastomeric sensor. In *ACM Transactions on Graphics (TOG)*, volume 30, page 46. ACM, 2011.

[18] Zhanat Kappassov, Daulet Baimukashev, Zharaskhan Kuanyshuly, Yerzhan Massalin, Arshat Urazbayev, and Huseyin Atakan Varol. Color-coded fiber-optic tactile sensor for an elastomeric robot skin. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2146–2152. IEEE, 2019.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Julien Letessier and François Bérard. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 119–122. ACM, 2004.

[22] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3988–3993. IEEE, 2014.

[23] Hongbin Liu, Juan Greco, Xiaojing Song, Joao Bimbo, Lakmal Seneviratne, and Kaspar Althoefer. Tactile image based contact shape recognition using neural network. In *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 138–143. IEEE, 2012.

[24] Asanterabi Kighoma Malima, Erol Özgür, and Müjdat Çetin. A fast algorithm for vision-based hand gesture recognition for robot control. *IEEE (Institute of Electrical and Electronics Engineers)*, 2006.

[25] Jérôme Martin, Vincent Devin, and James L Crowley. Active hand tracking. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 573–578. IEEE, 1998.

[26] Takashi Minato, Yuichiro Yoshikawa, Tomoyuki Noda, Shuhei Ikemoto, Hiroshi Ishiguro, and Minoru Asada. Cb2: A child robot with biomimetic body for cognitive developmental robotics. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 557–562. IEEE, 2007.

[27] Cristina Nuzzi, Simone Pasinetti, Matteo Lancini, Franco Docchio, and Giovanna Sansoni. Deep learning-based hand gesture recognition for collaborative robots. *IEEE Instrumentation & Measurement Magazine*, 22(2):44–51, 2019.

[28] Shijia Pan, Ceferino Gabriel Ramirez, Mostafa Mirshekari, Jonathon Fagert, Albert Jin Chung, Chih Chi Hu, John Paul Shen, Hae Young Noh, and Pei Zhang. Surfacevibe: vibration-based tap & swipe tracking on ubiquitous surfaces. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 197–208. IEEE, 2017.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct): 2825–2830, 2011.

[30] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Transfer learning for image classification with sparse prototype representations. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[31] Miguel A Salichs, Ramon Barber, Alaa M Khamis, María Malfaz, Javier F Gorostiza, Rakel Pacheco, Rafael Rivas, Ana Corrales, Elena Delgado, and David Garcia. Maggie: A robotic platform for human-robot social interaction. In *2006 IEEE Conference on Robotics, Automation and Mechatronics*, pages 1–7. IEEE, 2006.

[32] Siddharth Sanan, Michael H Ornstein, and Christopher G Atkeson. Physical human interaction for an inflatable manipulator. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 7401–7404. IEEE, 2011.

[33] Johannes Schöning, Jonathan Hook, Tom Bartindale, Dominik Schmidt, Patrick Oliver, Florian Echtler, Nima Motamedi, Peter Brandl, and Ulrich von Zadow. Building interactive multi-touch surfaces. In *Tabletops-Horizontal Interactive Displays*, pages 27–49. Springer, 2010.

[34] Takanori Shibata. Ubiquitous surface tactile sensor. In *IEEE Conference on Robotics and Automation, 2004. TExCRA Technical Exhibition Based.*, pages 5–6. IEEE, 2004.

[35] David Silvera-Tawil, David Rye, and Mari Velonaki. Interpretation of social touch on an artificial arm covered with an eit-based sensitive skin. *International Journal of Social Robotics*, 6(4):489–505, 2014.

[36] Walter Dan Stiehl and Cynthia Breazeal. Affective touch for robotic companions. In *International Conference on Affective Computing and Intelligent Interaction*, pages 747–754. Springer, 2005.

[37] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[38] Yoshiki Takeoka, Takashi Miyaki, and Jun Rekimoto. Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 91–94, 2010.

[39] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.

[40] D Silvera Tawil, David Rye, and Mari Velonaki. Improved eit drive patterns for a robotics sensitive skin. In *Proceeding of Australasian Conference on Robotics and Automation (ACRA), Sydney, Australia*, pages 2–4, 2009.

[41] David Silvera Tawil, David Rye, and Mari Velonaki. Touch modality interpretation for an eit-based sensitive skin. In *2011 IEEE International Conference on Robotics and Automation*, pages 3770–3776. IEEE, 2011.

[42] Kazuyoshi Wada and Takanori Shibata. Living with seal robots – its socio-psychological and physiological influences on the elderly at a care house. *IEEE transactions on robotics*, 23(5):972–980, 2007.

[43] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics*, 5(2):216–227, 2018.

[44] Andrew D Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76, 2004.

[45] Benjamin Winstone, Gareth Griffiths, Tony Pipe, Chris Melhuish, and Jonathon Rossiter. Tactip-tactile fingertip device, texture analysis through optical tracking of skin features. In *Conference on Biomimetic and Biohybrid Systems*, pages 323–334. Springer, 2013.

[46] Dan Xu, Yen-Lun Chen, Chuan Lin, Xin Kong, and Xinyu Wu. Real-time dynamic gesture recognition system based on depth perception for robot navigation. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 689–694. IEEE, 2012.

[47] Jie Yang and Alex Waibel. Tracking human faces in real-time. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School Of Computer Science, 1995.