



Pressure distribution classification and segmentation of human hands in contact with the robot body

The International Journal of
Robotics Research
1–20

© The Author(s) 2020



Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/0278364920907688

journals.sagepub.com/home/ijr



Alessandro Albini^{ID} and Giorgio Cannata

Abstract

This article deals with the problem of the recognition of human hand touch by a robot equipped with large area tactile sensors covering its body. This problem is relevant in the domain of physical human–robot interaction for discriminating between human and non-human contacts and to trigger and to drive cooperative tasks or robot motions, or to ensure a safe interaction. The underlying assumption used in this article is that voluntary physical interaction tasks involve hand touch over the robot body, and therefore the capability to recognize hand contacts is a key element to discriminate a purposeful human touch from other types of interaction. The proposed approach is based on a geometric transformation of the tactile data, formed by pressure measurements associated to a non-uniform cloud of 3D points (taxels) spread over a non-linear manifold corresponding to the robot body, into tactile images representing the contact pressure distribution in two dimensions. Tactile images can be processed using deep learning algorithms to recognize human hands and to compute the pressure distribution applied by the various hand segments: palm and single fingers. Experimental results, performed on a real robot covered with robot skin, show the effectiveness of the proposed methodology. Moreover, to evaluate its robustness, various types of failures have been simulated. A further analysis concerning the transferability of the system has been performed, considering contacts occurring on a different sensorized robot part.

Keywords

Tactile sensing, robot skin, human–robot interaction

1. Introduction

Human–robot interaction (HRI) has the goal of making possible the cooperation between humans and robots, in order to exploit the strengths of both *players* to accomplish complex tasks, that are otherwise difficult to tackle, or tedious and error prone. Towards this aim, and in order to ensure safe interaction, robots are expected to embed human-like sensing modalities such as vision, touch, speech, etc.

In the literature HRI has been largely based on vision systems, for example to recognize gestures (Li, 2012), to cooperate with robots in assembly tasks (Kimura et al., 1999), and to deal with collision detection problems (Ebert and Henrich, 2002).

Of course, when contacts occur, interaction control of the robot is required based on the capability of sensing the contact phenomena. To achieve this, force/torque sensors have been largely used in order to ensure safe physical HRI (pHRI), by detecting collisions (Haddadin et al., 2008) and ensuring robot compliant behavior in response to external

forces (Duchaine and Gosselin, 2007; Grunwald et al., 2003).

Bicchi et al. (1993) have shown that for a given robot geometry for contacts over *small* areas it is possible to reconstruct the interaction forces and the contact centroid location by processing lumped force/torque measurements. Although this method has been proven effective for object manipulation using robot hands, it can be hardly scaled in case of multiple contacts, or complex interactions expressed over large areas, which are phenomena expected to arise in tasks involving tight HRI.

Humans perceive contacts mostly through the skin; therefore, tactile sensors mimicking its functionality and

Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS), University of Genoa, Genoa, Italy

Corresponding author:

Alessandro Albini, Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS), University of Genoa, Via Opera Pia 13, 16145, Genoa, Italy.

Email: alessandro.albini@dibris.unige.it

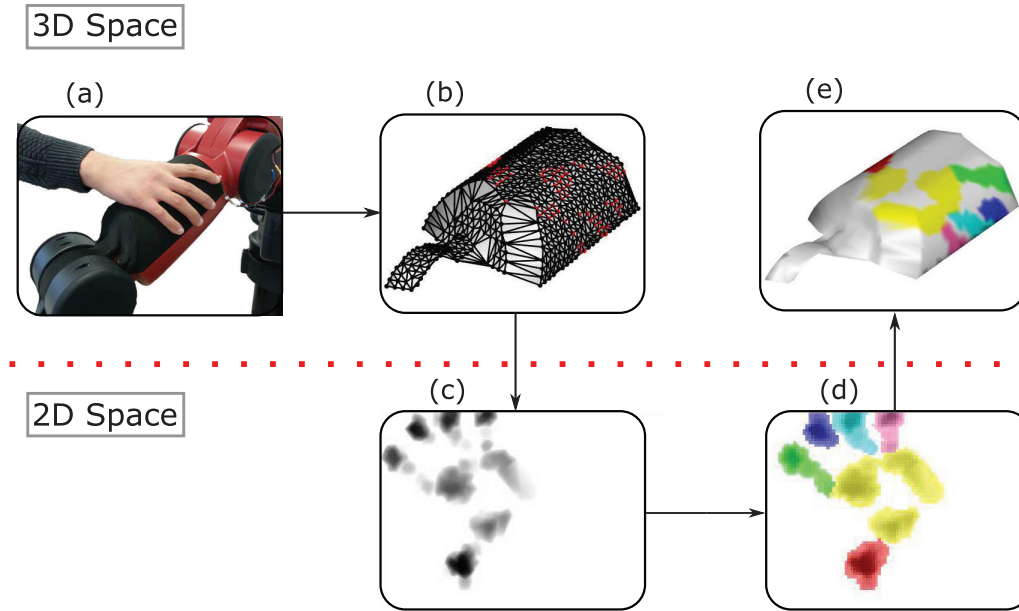


Fig. 1. Proposed approach: (a) a human is touching the robot arm using the hand; (b) the 3D contact measurements are mapped onto the mesh representing the robot body; (c) the robot skin measurements are transformed into an image and classified to recognize a human hand; (d) if it is, the parts of the hand are segmented; (e) the segmentation is back-projected onto the original 3D space.

integrated on the robot body are expected to provide additional information with respect to force/torque sensors. Large-area tactile sensors composed of different transducers (pressure, vibration, temperature, proximity, etc.), also referred to as *robot skin*, have been proposed in the past years by several authors (Cannata et al., 2008; Cheung and Lumelsky, 1989; Minato et al., 2007; Mittendorfer and Cheng, 2011; Mizuuchi et al., 2006; Mukai et al., 2008; Ohmura et al., 2006; Someya et al., 2004; Tawil et al., 2011; Um et al., 1998). Typically, robot skin sensors should make it possible to measure the contact pressure distribution applied on the robot body over an arbitrary area, thus opening new scenarios in pHRI, for control and for cognitive level processing, enabling the interpretation of physical contacts.

Usually, humans physically interact with objects, or with other people, hopefully in peaceful conditions, using their hands. Similarly, in HRI it can be expected that if an operator wants to physically interact with a robot, for example to teach a movement (Billard et al., 2008), a natural way to begin the cooperation would be touching or grasping one or more of its links. In fact, various vision-based HRI methods are based on the assumption that the hands are the main input for interacting with robots. Indeed, they address the problem of computing from images the placement of the fingers and of the palm of the human player (Liang et al., 2012; Raheja et al., 2011) in order to recognize gestures. In the pHRI domain, it can then be argued that when a person interacts using the hand, the contact distribution generated by each finger and by the palm, in terms of positions, areas, and relative applied pressures, could imply a specific type of interaction.

Therefore, according to what discussed so far, it is reasonable to assume that if a human is interacting with a robot

using their hand, the contact could be interpreted as a voluntary touch, performed to start a cooperation. Then, in order to engage an appropriate HRI task, the robot must be capable of discriminating whether the applied contact has been generated by a human and it should be capable of segmenting the measured pressure distribution associated with the various parts of the hand.

In this work, we present a method based on robot skin feedback measurements to:

- **recognize a human voluntary touch** performed using a single hand, with respect to a generic contact or collision;
- **segment the hand contact shape**, obtaining the pressure distribution applied by each part of the hand (fingers and palm) during the interaction.

As shown in Figure 1, the proposed approach consists of creating a *tactile image* of the contact distribution by performing a set of geometric transformations making it possible to obtain a planar 2D representation of the robot body. The main advantage of using this technique is that it allows state-of-the-art image processing techniques to be applied. As explained in detail in Section 4, the pressure distribution will be classified and segmented using machine learning techniques because the variabilities produced by a human touching a robot skin make the definition of interaction models hard. The novelty of the proposed approach is that the tactile images are generated from robot skin measurements, where pressure sensors are distributed in a *non-uniform* way over a complex non-planar *2D manifold* (i.e., the robot body). Indeed, whereas tactile images have been used to process data in the case of small-scale planar tactile

sensors, to the best of the authors' knowledge, tactile images originated from a non-regular large-area distribution of tactile sensors have been first proposed in Albini et al. (2017b): this article completes and extends those results. In particular, beyond the original problem of the human hand contact recognition (Albini et al., 2017b), this article also investigates the problem of the human hand contact segmentation. Furthermore, because robot skin is prone to failure owing to its nature, a robustness analysis of the performance of the classification and segmentation models against different types of tactile failures has been performed. Finally, an analysis of the transferability of the hand recognition system has been experimentally performed by testing the proposed method on tactile data originated from contacts occurred on a completely different robot part.

This article is organized as follows. Section 2 provides a review of the literature: first the use of tactile sensors in pHRI is discussed; second the techniques related to contact shape processing are analyzed, discussing the differences and the improvements proposed in this article. Sections 3 and 4 describe the process of computing tactile images from robot skin feedback and the specific problems related to the processing of human hand contact shapes, respectively. Sections 5 and 6 describe the machine learning-based models employed for human hand recognition and segmentation. In Section 7, the experimental setup and the data collection procedure are detailed. The experimental results to assess the performance of the proposed method are discussed in Section 8. In Section 9, additional experiments are presented to analyze: (i) the robustness of the system with respect to hardware failures and changes in the spatial resolution; (ii) the transferability of the system, by testing it on a different sensorized robot part. Conclusions follow in Section 10.

2. Related work

Within the scope of this article, the role of tactile sensors has been studied with respect to two different domains of application. The first is related to HRI and the second to contact shape processing and classification.

2.1. Tactile sensors in pHRI

Tactile sensors measurements have been used in the context of HRI in order to implement touch-based control strategies.

Wosch and Feiten (2002) showed that patches of pressure sensors integrated on a robot link allow human operators to guide a robot arm. The pressure readings are translated into motion vectors used for controlling the arm position. Similarly, Schmidt et al. (2006) used an array of capacitive-based pressure sensors mounted on a robot gripper to implement a control strategy allowing the robot to adapt its posture in response to the force applied by a human operator.

Frigola et al. (2006) implemented a compliant behavior in a robot arm exploiting the feedback of a force-sensitive bumper skin. Leboutet et al. (2016) achieved whole-robot-body compliance by using a technique based on hierarchical force propagation exploiting force feedback provided by an artificial skin. Albini et al. (2017a) proposed a touch-triggered task-based control method using robot skin tactile feedback allowing a human operator to physically drive robot motions in Cartesian or joint space.

Tactile sensors have been also used to recognize different *touch modalities*, namely actions (e.g., *pat*, *push*, etc.) performed by human subjects using the hand. The general approach is similar in most of the techniques proposed in the literature: a set of features is extracted and classified using supervised machine learning algorithms (e.g., Silvera-Tawil et al., 2015), the main differences among the various solutions being the number of modalities classified and the training methodologies adopted. In particular, Naya et al. (1999) used a *k*-neighbor algorithm to classify 5 touch modalities, based on data collected in experiments involving 11 users. A neural network has been considered by Stiehl and Breazeal (2005) in order to classify a set of eight interactions performed by a single subject. Tawil et al. (2012) used the *LoogitBoost* algorithm (Friedman et al., 1998) to recognize 9 touch modalities acquired from 40 subjects. Finally, Kaboli et al. (2015) implemented a support vector machine (SVM) to recognize nine touch modalities using a multimodal robot skin providing pressure, acceleration, and proximity measurements.

In all the works discussed above it is implicitly assumed that a person is interacting with the robot: namely, all the contacts used for the classification have been generated by humans. Therefore, they all have not been addressing the possibility of discriminating human touch from other possible types of contacts. We show in this article that such a discrimination can be achieved by analyzing the shape of the contact pressure distribution. A review of the methods and techniques for contact shape processing is presented in the following.

2.2. Contact shape processing and classification with tactile images

In applications requiring the processing and classification the *contact shape*, it is common to convert the pressure data distribution into a tactile image, which is a representation where the intensity of each pixel corresponds to a pressure value. The advantage is obviously that tactile images can be processed or classified using state-of-the-art image processing techniques.

Schneider et al. (2009) used a small pressure array integrated onto a robot fingertip to actively touch objects of interest and the resulting tactile images were classified using a bag of visual words (BoVW) model. Liu et al. (2012b) showed that tactile images generated from a fingertip can be used to classify in real-time primitive shapes and

poses of the contact. Liu et al. (2012a) covered a robot hand with small planar tactile patches mapping the whole pressure readings onto a single image. Finally, they trained a neural network to classify a set of grasped objects. Cao et al. (2016) used a stream of tactile images obtained during a grasping task to classify 10 different objects using a convolutional neural network (CNN). Gandarias et al. (2018) proposed an approach where a high-resolution patch of pressure sensors integrated on a gripper is used to classify the tactile images generated by objects, human limbs, and fingers through a CNN.

In addition to the use of robot hands, other approaches employ a rectangular patch of tactile sensors mounted on the robot end-effector. Pezzementi et al. (2011) proposed to obtain a set of tactile images generated from a sequence of contacts and used a BoVW model for object recognition. A similar approach has been considered by Luo et al. (2015b) in order to classify a set of objects using an innovative *tactile SIFT* descriptor (a specialization of the scale-invariant feature transform (SIFT) algorithm originally developed for image data processing). The extracted features are then classified using the visual bag of words algorithm producing very good classification results. Taking advantage of the similarity between tactile and visual images, the same authors proposed algorithms to merge tactile and visual feedback for object localization and classification (Liu et al., 2017; Luo et al., 2015a). The combination of tactile and visual feedback has also been exploited by Yang and Lepora (2017) to implement an object exploration strategy.

Therefore, it appears clear from the previous discussions that tactile images have been proved to be a powerful tool for classifying tactile data, although in most of the cases they have been generated from planar tactile patches containing sensors distributed on a regular grid with uniform spatial resolution and generally covering a small area.

3. Tactile image formation from distributed tactile sensors measurements

In this section, the problem of generating a tactile image from a contact distributed on the robot body is addressed. The proposed technique makes possible to create a picture of the contact with minimal distortion with respect to the original 3D shape.

3.1. Map the robot body onto a flat representation

It is assumed to have a robot link covered with robot skin (see Figure 2(a) as an example). The robot skin is here intended as a set of N distributed pressure transducers called *taxels*.

The position and the response of each taxel to a given pressure stimulus on the robot body are assumed to be known, possibly as the outcome of a calibration procedure. Then it is possible to define the set $T = \{\mathbf{t}_1, \dots, \mathbf{t}_N\}$, where

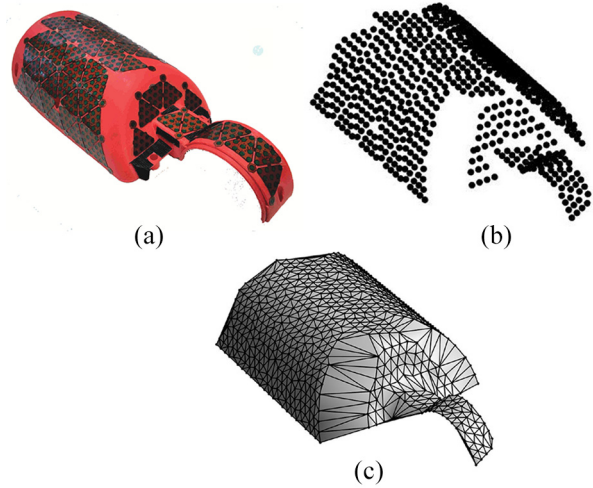


Fig. 2. Steps for constructing the 3D mesh S^* . (a) Real robot link covered with robot skin. (b) Placement of the taxels obtained from the spatial calibration of the skin. (c) The mesh S^* approximating the robot body shape S .

the element $\mathbf{t}_i \in \mathbb{R}^3$ represents the 3D position of the i th taxel; the set T can be intended as a sort of *point cloud* where each taxel position \mathbf{t}_i is referred with respect to the reference frame of the sensorized robot link (see Figure 2(b)).

A *Delaunay triangulation* (Fortune, 1997) applied to T , allows to us define a list of topological relations F between adjacent taxels, thus creating a 3D mesh $S^* = (T, F)$, representing a piecewise linear approximation of the robot link shape S (see Figure 2(c)).

As proposed by Cannata et al. (2010), the idea is to exploit the *surface parameterization theory* (Desbrun et al., 2002) to transform the mesh S^* into a 2D *flattened* representation of the robot body, thus allowing to preserve sensor locations, displacements, density, and proximity relationships among the sensors. Formally, the flattening allows us to define a piecewise linear mapping $\Psi : S \rightarrow M$ between the robot body surface S and an isomorphic 2D (flat) surface M , also called a *tactile map* in the following, defined by a mesh of points $M^* = (\{\mathbf{m}_1, \dots, \mathbf{m}_N\}, F)$ where the elements $\mathbf{m}_i \in \mathbb{R}^2$ best preserve the properties of the mesh S^* minimizing the distortions from three to two dimensions. Therefore, for each \mathbf{t}_i , a corresponding \mathbf{m}_i exists such that $\mathbf{t}_i = \Psi^{-1}(\mathbf{m}_i)$. An example of the flattening transformation applied to the mesh in Figure 2(c) is shown in Figure 3(a).

The method described above refers to a class of robot skin systems composed of discrete taxels rigidly attached to the robot links. There are several examples of technologies corresponding to this assumption (e.g., Cheung and Lumelsky, 1989; Minato et al., 2007; Mittendorf and Cheng, 2011; Mizuuchi et al., 2006; Mukai et al., 2008; Ohmura et al., 2006; Schmitz et al., 2011).

Remark 1. *Conceptually the method could also be applied to other robot skin technologies not based on discrete taxel sensing, provided that the geometry of the sensor surface is known and that the pressure at discrete points can be*

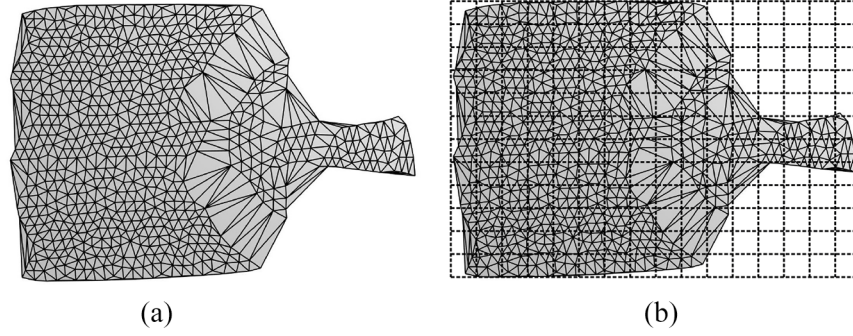


Fig. 3. Steps for constructing the tactile image from a 2D mesh with a non-uniform placement of the taxels. (a) Robot tactile map M^* , obtained by flattening the 3D mesh S^* . (b) A regular grid superimposed on M^* . Barycentric interpolation allows the computation of the pressure values corresponding to the nodes of the grid.

computed or estimated. One example of these types of tactile systems is that based on EIT technology (Tawil et al., 2011).

Remark 2. It is also worth noting that the computation of the map Ψ can be performed off-line for contacts expressed on a single link. Then, it does not pose significant problems for real-time computations because, in practice, the map Ψ is implemented as a look-up table. In the case of more complex type of contacts involving more than one link, the flattening should be computed, in principle, at each given robot posture. These computational aspects are beyond the scope of this article; however, suboptimal flattening procedures addressing the problem of the relative displacement of the taxels caused by robot motion has been preliminary addressed in Albini and Cannata (2018).

3.2. Tactile image creation

The tactile map M^* is a 2D entity representing the non-uniform planar displacement of the taxels. In order to generate a tactile image, M^* must be re-sampled. This is done by superimposing a regular grid with R rows and C columns on the tactile map M^* , as shown in Figure 3(b). The position of the grid point corresponding to row r and column c is defined as \mathbf{x}_{rc} .

During a contact, the robot skin senses the applied pressure generating a set of measurements $P = \{p_1, p_2, \dots, p_N\}$, where $p_i \in \mathbb{R}$ is the measurement of the i th taxel. Figure 4(b) represents the discrete pressure distribution of the contact at a given time instant, obtained by associating the tactile measurements P to the mesh S^* . Similarly, P can be mapped on M^* generating a discrete *pressure map* (see Figure 4(c)).

Remark 3. In Figure 4(b) and (c) all the taxels involved in the contact are marked as red dots for clarity of visualization. The actual sensor taxel response is assumed to be continuous and not binary (as better detailed in Section 7.1).

In order to compute the tactile image (see Figure 4(d)), for each point of the grid \mathbf{x}_{rc} that lies in the triangle defined

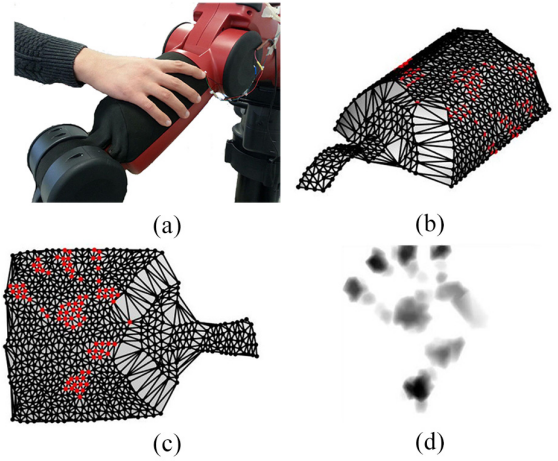


Fig. 4. Steps to obtain a tactile image. (a) Example of a physical contact of a hand on the robot forearm. (b) Pressure measurements mapped onto the mesh S^* (actual intensity values not shown for clarity). (c) Pressure measurements applied on the tactile map M^* (actual intensity values not shown for clarity). (d) Resulting tactile image of the contact obtained with a grid of 247×362 pixels.

by $(\mathbf{m}_j, \mathbf{m}_k, \mathbf{m}_h)$, a pressure value K_{rc} is computed, using the *barycentric interpolation*:

$$K_{rc} = \frac{(A_{kj}p_h + A_{hj}p_k + A_{hk}p_j)}{A}$$

where p_j , p_k , and p_h are the pressure values of the taxels associated with \mathbf{m}_j , \mathbf{m}_k , \mathbf{m}_h , whereas A , A_{kj} , A_{hj} , and A_{hk} are the areas of the triangles defined by the vertices $(\mathbf{m}_j, \mathbf{m}_k, \mathbf{m}_h)$, $(\mathbf{m}_j, \mathbf{m}_k, \mathbf{x}_{rc})$, $(\mathbf{m}_h, \mathbf{m}_j, \mathbf{x}_{rc})$, and $(\mathbf{m}_h, \mathbf{m}_k, \mathbf{x}_{rc})$, respectively (see Figure 5).

Here K_{rc} are the elements of a matrix \mathbf{K} that can be converted into a classical *grayscale image*, by scaling each K_{rc} value into a grayscale level I_{rc} , with the following formula:

$$I_{rc} = 255 \left\lfloor \frac{K_{rc}}{\max(p_i)} \right\rfloor$$

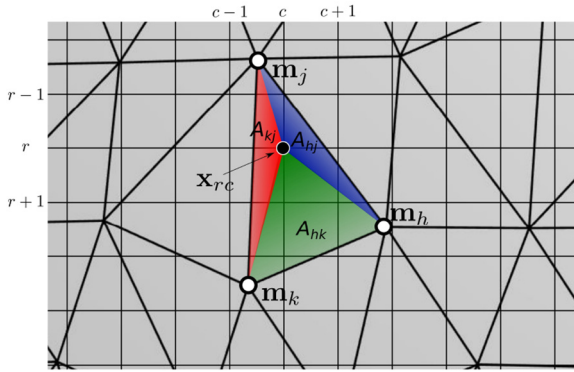


Fig. 5. Interpolation with the pressure values of nearby vertices.

where $\lfloor \cdot \rfloor$ is the floor function and $i = \{1, 2, \dots, N\}$.

The conversion described previously generates a tactile image normalized with respect to the maximum value measured in the current contact. This is motivated by the fact that in this work we focus only on the shape profile generated during the contact. The normalization of \mathbf{K} allows to highlight the contact shape, making the classification and segmentation of the pressure distribution independent from the magnitude of the applied contact pressure. However, it is worth noting that the normalization above is used for the tactile image generation only, while the actual pressure exerted is known from P (or in the interpolated form \mathbf{K}).

4. Tactile images from human hand contacts

Some examples of human hand tactile images generated with the discussed procedure are shown in Figure 6. As it

can be seen, in some images it is possible to identify the shape of the human hand, while other pictures (e.g., Figure 6(b), (d), (f), and (g)), can be easily confused with the non-hand contacts in Figure 7. However, it is quite evident that the contact shape can vary significantly even in the images where the hand is visible. For example, Figure 6(l) clearly shows the human hand shape, whereas others just show a portion of the hand or possibly only the fingertips. This is due to various factors linked to the geometry of the robot skin and to the characteristic of the interaction.

Aspects related to robot skin

- Unlike cameras, the spatial resolution of the tactile elements composing the skin can be non-uniform. Therefore, there could be areas poorly or even not sensorized at all that could produce *holes* (loss of information) in the resulting tactile image.
- The flattening operation introduces distortions dependent on the “complexity” of the robot body shape. This implies that the similar contacts applied in different positions can produce slightly different 2D tactile images. Examples of this fact are given in Figure 6(h) and (k) where the fingers appear to be bent, or in Figure 6(a) and (f) where the distortions are more evident.

Aspects related to human interaction

- The tactile images are characterized by the type of interaction: for example, while pushing away the robot arm requires the whole hand, pulling the same part mainly involves the fingertips; moreover, in

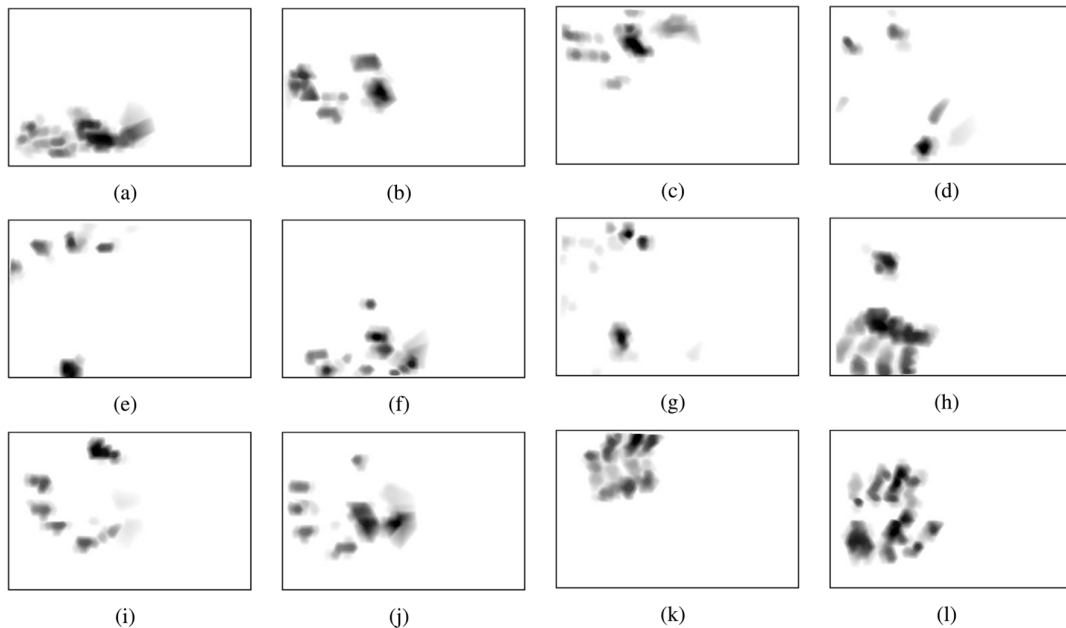


Fig. 6. Examples of tactile images generated by human subjects during different interactions with the robot. Some fingertips seem to be cut (e.g., Figure 6(h)) because the person did not fully touch the sensorized area.

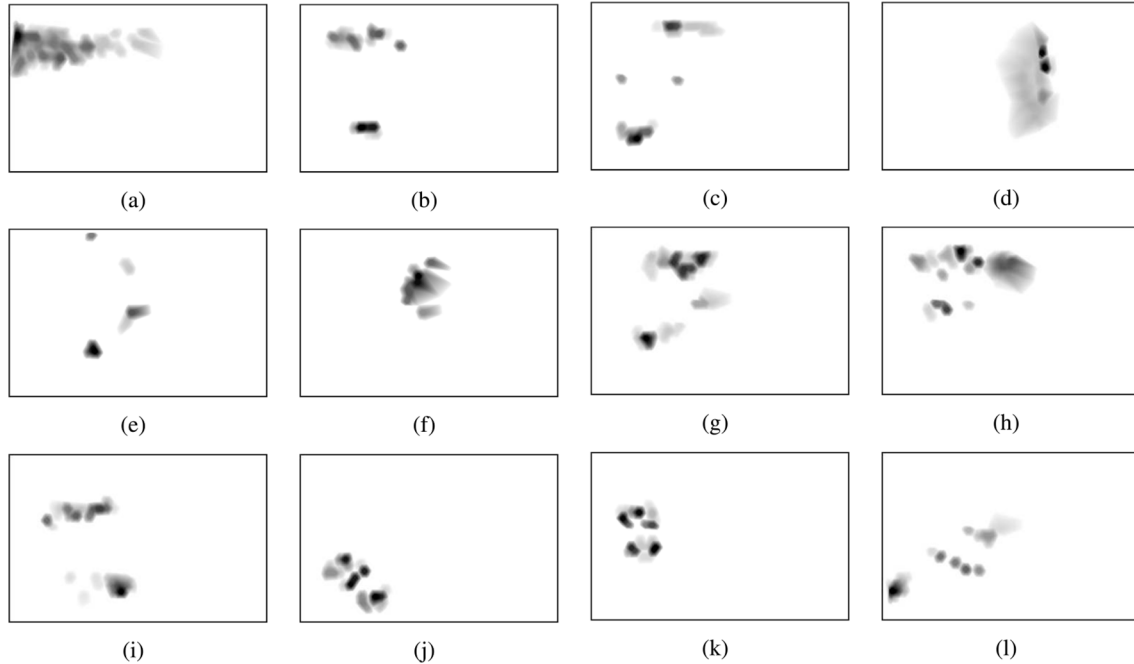


Fig. 7. Examples of tactile images not generated by hand contacts.

some actions not all the fingers or the palm are involved (see Figure 6(e) and (k)).

- Depending on the human operator physical characteristics (e.g., height, size of the hands, strength, etc.) and their relative posture with respect to the robot, each subject will interact with the robot body with different intensities or configurations of the hand; for example, Figure 6(i) and (j) represent a similar contact geometry expressed with different pressure distributions.

Owing to these variabilities, it is hard if not impossible to define a general model of a human hand in contact with a robot body.

For this reason, because our goal is to classify and segment the pressure distribution, it appears reasonable to use machine-learning-based techniques. In particular, *supervised* methods have been considered.

5. Hand classification

In order to recognize whether the contact distribution is generated by a human hand, the corresponding tactile image is classified using machine learning techniques.

CNNs for image classification outperformed previous approaches (Krizhevsky et al., 2012), proving their robustness against image variations such as scale and rotation (Farfadi et al., 2015). Moreover, they have been successfully employed to recognize hand gestures in real time (Kim et al., 2008; Lin et al., 2014; Nagi et al., 2011)

and in tasks of tactile objects classification (Cao et al., 2016).

In this work a CNN classifier trained from scratch for recognizing the human hand touch, referred in the following as **HandsNet**, is proposed. Then, because this CNN architecture is not specific for tactile measurements, but it works on images, its performance will be compared with a pre-trained model (Yosinski et al., 2014). Furthermore, because several works discussed in Section 2.2 rely on the BoVW model for classifying tactile images, also the performance of this model is tested.

Table 1 shows the layers of the **HandsNet** model. The first part is composed of four stacked convolutional blocks, each containing three layers: a convolutional layer with padding and stride equal to one, a batch normalization layer, and, finally, a threshold operation performed through a rectified linear unit (ReLU) layer (Goodfellow et al., 2016). Then the output is downsampled with a 2×2 MaxPool filter with stride 2 before being further processed.

The differences among the four blocks are in the number of filters of the convolutional layers and in the size of the kernels. According to Goodfellow et al. (2016), the depth of the network has been selected by increasing the number of layers and evaluating the accuracy on the training set, until a satisfactory performance has been obtained. The output of the last max pooling operator is sent as an input to a fully connected layer composed of 64 neurons (fc_1 in Table 1). Two further fully connected layers containing 32 and 2 neurons, respectively, follow. Finally, the output is a

Table 1. Structure of **HandsNet**. The nomenclature $conv_i$ refers to a computational block formed by a convolutional layer followed by a batch normalization and, finally, by ReLU.

Layer	Shape
conv_1	$32 \times 7 \times 7$
max_pool_1	2×2
dropout_1 (10%)	—
conv_2	$64 \times 5 \times 5$
max_pool_2	2×2
dropout_2 (20%)	—
conv_3	$128 \times 3 \times 3$
max_pool_3	2×2
conv_4	$256 \times 3 \times 3$
max_pool_4	2×2
fc_1	64
dropout (60%)	—
fc_2	32
dropout (50%)	—
fc_3	2
softmax	2

two-way softmax unit computing a probability distribution over two classes: *hand* and *non-hand*. In order to reduce the overfitting, dropout layers have been inserted, by choosing their probabilities according to Park and Kwak (2017) who suggested applying a low drop rate in the initial layers (usually less than 0.5).

The classification performance of **HandsNet** has been compared with other state-of-the-art models used in image classification. Focusing on pre-trained CNNs, there are mainly two ways to adapt a model to a particular problem. As the initial layers of the network are able to extract generic features (Yosinski et al., 2014), one possible solution is to remove the classification layers and to use the network as a feature extractor. Once the features are computed for the new dataset, they can be used to train a new classifier (e.g., a SVM).

The other approach is the fine-tuning, consisting of replacing the classification layer with a new one having the appropriate number of classes and then retraining the network. During this phase, the strategy is to use a very small learning rate to update the weights of the initial layers. In contrast, a higher learning rate is applied to train the final layers, by adapting them to the new data.

Both methods have been considered in this study applied to the VGG16 model presented in Simonyan and Zisserman (2014). This model is pre-trained on the ImageNet dataset (Deng et al., 2009), and it has been proved to be a very good choice to initialize a classifier or to be used as a feature extractor (Guo et al., 2016).

Finally, the last model considered is the BoVW model, already exploited for tactile image classification.

To summarize, the four following models will be evaluated and compared.

- **HandsNet**: the model having the structure described in Table 1.
- **VGG16 + SVM**: the features are extracted with the pre-trained VGG16 and classified using a linear SVM.
- **VGG16 + ft**: fine tuning on the VGG16 pre-trained model.
- **BoVW**: BoVW model trained with SIFT features (Lowe, 2004).

The loss function and the hyper-parameters used during the training phase are detailed in the Appendix.

6. Hand segmentation

The goal of this section is to describe how to segment the pressure distribution applied by a human hand, in order to identify the fingers and the palm area. As tactile images are used, this task can be seen as a problem of *semantic segmentation*. In addition in this case, an approach using deep learning has been considered. Indeed, the segmentation of tactile images is specific, because the number of classes could vary depending on the type of contact (e.g., the number of fingers touching the robot body could change). Furthermore, the regions composing a part of the hand could be not connected, as for the case of the palm contact in Figure 6(l). Therefore, the classical techniques often referred in the literature (such as k -means, watershed, thresholds, etc.) do not appear to be suitable in this context (Dhanachandra et al., 2015; Grau et al., 2004; Morar et al., 2012).

Modern approaches presented in the past few years, dealing with the problem of semantic segmentation, rely on deep networks performing classification tasks (Guo et al., 2018), where a label is associated with each pixel instead of the whole image. In this article, two models have been considered: the **SegNet** (Badrinarayanan et al., 2017) and **FCN** (Long et al., 2015). Both are widely applied in the literature, representing the state of the art in semantic segmentation (Garcia-Garcia et al., 2018).

Deep networks performing a pixel-wise classification require a large amount of data to be trained from scratch. Although we collected a dataset of human hand contacts (as detailed in the next section), the pixel-wise classification of the whole dataset is a time-consuming operation. For this reason the convolutional layers of both models are initialized with the weights of a VGG16 model trained on ImageNet. In this way, the network can be trained using less data, thus requiring just a portion of the whole dataset to be labeled.

The two models have been trained in order to segment and recognize the following six classes: *Thumb*, *Index*, *Middle*, *Ring*, *Pinkie*, and *Palm*. The training details are reported in the Appendix.

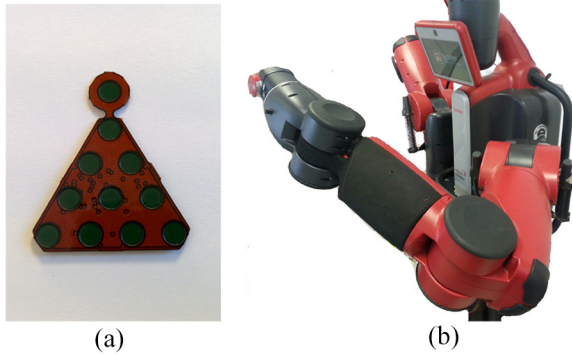


Fig. 8. Experimental setup. (a) Triangular module of the robot skin. The diameter of each taxel is 3.5 mm, with a pitch of 8 mm among nearby taxels. (b) Sensorized link mounted on the real robot and covered with a conductive ground plane.

7. Dataset

In this section, the robot skin technology and the procedure used to collect a dataset for training the machine learning models are described.

7.1. Experimental setup

The robot skin used in this work is an engineered version of the technology presented in Schmitz et al. (2011). In this new version the thickness of the dielectric has been reduced to 0.5 mm in order to improve the sensitivity of the sensor. The skin is composed of interconnected modules forming a network of sensors. Each single module (shown in Figure 8(a)) is implemented with a flexible PCB and contains 11 capacitive pressure transducers. A capacitance to digital converter embedded on each module provides, for each taxel, a response in the range 0–65,535.

As shown in Figure 2(a), the skin has been integrated on a Baxter robot, covering the upper part of the forearm with 768 pressure sensors. The final experimental setup is shown in Figure 8(b), where the forearm is mounted on the Baxter and covered with a black conductive fabric used as a ground plane.

7.2. Data collection

The dataset has been collected performing an experiment which involved voluntary human subjects. The experiment has been designed in order to capture the variabilities discussed in Section 4. The people were asked to interact with the robot arm performing the following actions:

1. grasp the forearm;
2. grasp and torque the forearm clockwise (i.e., a twist with respect to the forearm axis);
3. grasp and torque the forearm counter-clockwise;
4. push the forearm to the left;
5. push the forearm to the right;
6. push away the forearm;
7. pull the forearm.

Each action has been repeated twice in two different positions of the robot arm (see Figure 9). Each person interacted with the robot without any constraint related to the hand posture and intensity of the touch. After that, for five repetitions, the user moved the robot arm to a different configuration, performing one interaction of the list. In this phase, the arm position, the relative posture with respect to the robot, and the interaction type have been chosen by the user.

Throughout the whole experiment, the robot is commanded to maintain its pose and the entire interaction has been recorded. Each interaction produced a sequence of samples consisting of sensors measurements collected with a sampling time of 0.1 seconds. From this sequence, the sample with the highest number of taxels activated by the contact is selected to generate a single tactile image as described in Section 3. The tactile images have been generated using a regular grid with a step size of 1 mm. The robot tactile map (see Figure 3(a)) has a dimension of 247 mm \times 362 mm, so the corresponding tactile image is composed of 247 \times 362 pixels. Finally, in order to reduce the noise and further highlight the contact shape, an erosion followed by a dilatation of the image have been performed (Beyerer et al., 2016), using a circular structural element with two and four pixels of radius, respectively.

The experimental procedure discussed previously is the same followed in Albini et al. (2017b). The difference is that the number of people involved in the experiment has been increased from 43 to 90. The subjects have different gender (66.67% male, 33.33% female), handedness (77.78% right, 22.22% left), and biometric characteristics (Table 2). At the end of the data collection, 1,710 tactile images of hands have been acquired.

In order to train the models described in Section 5, the dataset has been completed by adding 1,820 *non-hand* images produced from contacts with other human limbs or generic objects. Contacts with objects have been collected by the authors over time by touching the robot on the sensorized area with objects having different properties such as shape, size, material (e.g., plastic, metal, etc.), and softness. The contacts with human body parts (e.g., torso, arm, forearm, shoulder, back) have been collected both by the authors and by the subjects involved in the experiment without using a formal protocol. In particular, all the users have been asked to touch the robot five times with different body parts other than the hand. In summary, about 35% non-hand images have been created from contacts with body parts and the remainder from contacts with objects.

Some examples are shown in Figure 7. As an outcome, the dataset used to train the classifiers in Section 5 is composed of 3,530 tactile images. The dataset has been split into a training set (70%) and a test set (30%). In order to evaluate the classifiers on previously unseen human subjects, the test set has been created containing images generated from subjects not included in the training set.

The semantic segmentation models described in Section 6 require pixel-wise labeled tactile images as ground truth.



Fig. 9. Two different positions taken by a human during the experiments: in front of the robot (a) and on its side (b).

Table 2. Summary of the characteristics of the subjects involved in the experiment. The hand length is measured from the wrist to the tip of the middle finger.

	Hand length	Age	Weight	Height
Min	15 cm	20	48 kg	154 cm
Max	22 cm	59	105 kg	194 cm
Mean	18 cm	26	70 kg	178 cm

According to the discussion in Section 6, the initialization with pre-trained weights allowed only a fraction of the whole dataset to be used. In particular, 350 samples have been picked from the whole dataset of human hands and labeled pixel by pixel. The distribution of classes is shown in Figure 10. In addition for this task, the dataset has been split into a training set (70%) and a test set (30%).

Both datasets, for the classification and segmentation tasks, are provided as supplementary material.

8. Experimental Results

This section reports the experimental results obtained with the models in Sections 5 and 6 using the datasets acquired as discussed in Section 7. The models have been trained on Matlab running on a server equipped with two Intel Xeon E5 CPUs and two Nvidia P100 GPUs with 16 GB of RAM each. For each model, a set of hyper-parameters has been selected and tuned. Details about the training and tuning procedures are reported in the Appendix.

8.1. Human hand touch classification

The models trained with the parameters described in the Appendix are evaluated on the test set. The results are given in Table 3 where the mean accuracy and the classification times are reported.

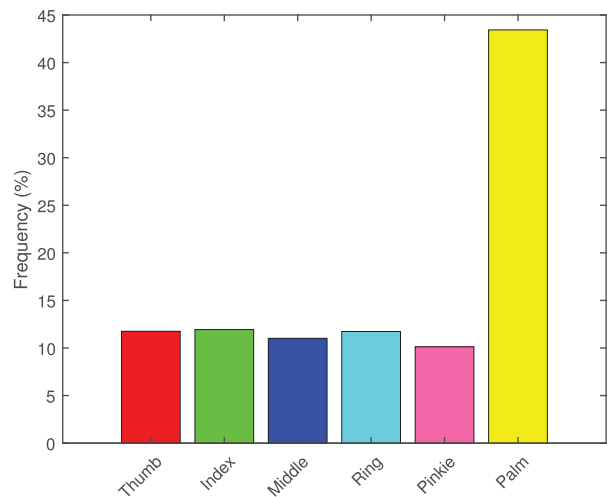


Fig. 10. Histogram representing the average frequency of pixels for segmented class. The colors shown in the histogram are also used in the following to identify the segments in the tactile images.

A more detailed analysis about the results obtained on the test set is given in Tables 4–7, representing the confusion matrices of the models.

It can be seen that **HandsNet** performs slightly better than **VGG16 + ft**. The difference in terms of accuracy is larger than 1% and it is faster with respect to **VGG16 + ft**. It is worth noting that the model **VGG16 + SVM** obtained good results in terms of accuracy and time, having only a single hyper-parameter to tune (see Appendix), whereas the **BoVW** produced lower performance with respect to the other models.

An example of tactile images misclassified by the **HandsNet** model is given in Figure 11, whereas the full list of tactile images classified correctly and misclassified for each model can be found in the provided supplementary material.

Table 3. Performance of the models. For each model, the mean accuracy on the test set and the time for classifying one tactile image have been computed.

	Accuracy	Time (ms)
HandsNet	97.81%	12.6
VGG16 + SVM	95.40%	14.4
VGG16 + ft	96.69%	27.5
BoVW	94.03%	17.6

Table 4. Confusion matrix of the **HandsNet** model applied on the test set. The mean accuracy is 97.81%.

	Hand	Non-hand
Hand	96.88%	1.28%
Non-hand	3.12%	98.72%

Table 5. Confusion matrix of the **VGG16 + SVM** model applied on the test set. The mean accuracy is 95.40%.

	Hand	Non-hand
Hand	96.49%	5.69%
Non-hand	3.51%	94.31%

Table 6. Confusion matrix of the **VGG16 + ft** model applied on the test set. The mean accuracy is 96.75%.

	Hand	Non-hand
Hand	98.64%	5.14%
Non-hand	1.36%	94.86%

Table 7. Confusion matrix of **BoVW** classifier applied on the test set. The mean accuracy is 94.03%.

	Hand	Non-hand
Hand	96.49%	8.44%
Non-hand	3.51%	91.56%

8.2. Human hand touch segmentation

To evaluate the models described in Section 6, the four metrics discussed in Long et al. (2015) have been considered. The first is the *pixel accuracy Acc*, which evaluates the percentage of correctly classified pixels without considering their classes. The second is the *pixel mean accuracy mAcc*, i.e., the percentage of correctly predicted pixels for each class, averaged over the classes. The third metric is the *mean intersection over union mIoU*, which computes how well the sets of predicted classes overlap the ground

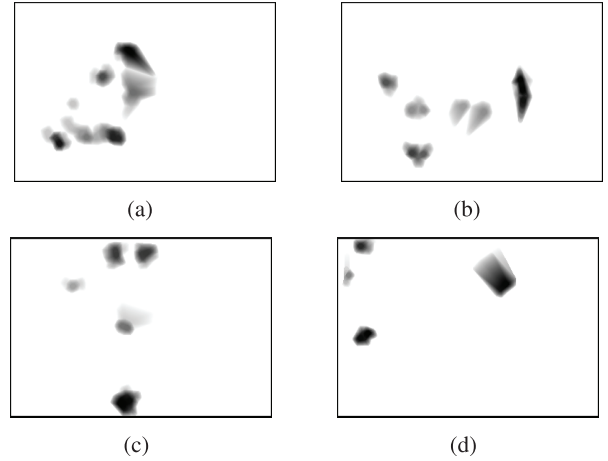


Fig. 11. Examples of tactile images misclassified by HandsNet; (a) and (b) non-human hand contacts classified as hands; (c) and (d) human hand contacts classified as non-hands.

truth. Finally, owing to the presence of imbalances in the dataset (see Figure 10), the *frequency weighted intersection over union fwIoU* has also been considered, i.e., a weighted version of the mIoU that takes into account the appearance frequency of each class.

Table 8 reports the scores obtained on the test set for each metric. **SegNet** model outperforms **FCN** providing also a lower inference time. The confusion matrices in Table 9 and 10 give detailed information about the pixel accuracy for each class. A comparative example between the two models is shown in Figure 12.

Focusing on **SegNet**, Figure 13(a)–(j) show a set of segmented tactile images (first row), along with the misclassified pixels (second row). As it can be seen, the network is able to correctly create the clusters under different conditions. For example in Figure 13(a) and (b) almost the whole hand is in contact with the robot body. In contrast, Figure 13(c), (d), and (e) show contacts where the fingers or palm are partially or completely not involved.

The network can also correctly segment fingers composed of non-connected regions as visible in Figure 13(f) and (g), or when the fingers are bent owing to the distortions introduced by the flattening (see Figure 13(h)). Figure 13(i) and (j) show instead two examples of poorly segmented tactile images with a mean pixel accuracy lower than 80%. The full list of images segmented using both models is included as supplemental material.

9. Robustness and transferability analysis

Owing to repeated physical contacts, the elements composing a robot skin are prone to failures. The complexity and the costs of the system could make it difficult or infeasible to replace a damaged part. Therefore, an analysis of the robustness of the proposed method is performed in the following, considering an increasing number of faulty tactile elements.

Table 8. Metrics evaluated for both models on the test set.

	Acc	mAcc	mIoU	fwIoU	Time (ms)
SegNet	93.37%	93.05%	89.17%	90.53%	63.37
FCN	88.82%	85.69%	80.14%	83.06%	75.13

Table 9. Confusion matrix of the **SegNet** model fed with the test set.

	Thumb	Index	Middle	Ring	Pinkie	Palm
Thumb	95.05%	0%	0%	0%	0.04%	0.92%
Index	0%	92.85%	2.59%	0.60%	2.10%	0.92%
Middle	0%	1.29%	90.34%	3.88%	0.25%	0.10%
Ring	0%	0.31%	5.07%	92.08%	1.03%	0.42%
Pinkie	0.38%	2.21 %	0.43%	2.56%	91.34%	1.02%
Palm	4.47%	3.33%	1.57%	0.88%	5.20%	96.61%

Table 10. Confusion matrix of the **FCN** model fed with the test set.

	Thumb	Index	Middle	Ring	Pinkie	Palm
Thumb	91.24%	0.75%	0.31%	0.23%	1.12%	0.90%
Index	0.81%	83.95%	3.31%	0.40%	3.51%	0.98%
Middle	0%	3.09%	78.41%	5.92%	1.30%	0.61%
Ring	0%	0.79%	9.91%	84.47%	1.48%	0.61%
Pinkie	0.6%	3.12%	1.62%	2.41%	80.10%	0.93%
Palm	7.88%	8.28%	6.42%	6.55%	12.47%	95.96%

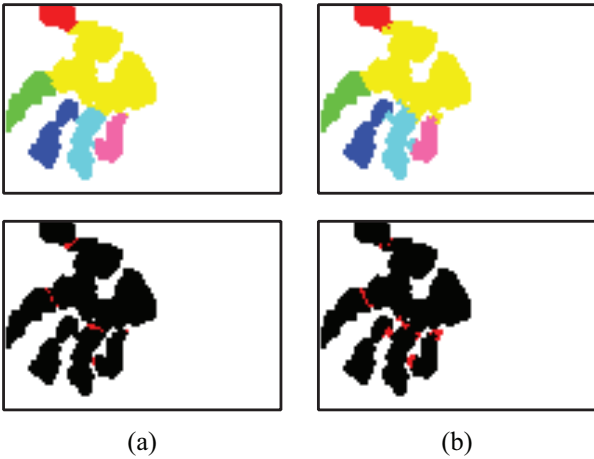


Fig. 12. Segmentation performed by SegNet and FCN on the same tactile image. (a) SegNet output mAcc: 98.77%. (b) FCN output mAcc: 94.86%. The first line shows the models output. The colors of the various segments are the same as used in Figure 10. The second line shows the tactile image in binary scale with red pixels corresponding to misclassified regions.

In particular, two different types of failures have been considered. In the first case, it is assumed that one or more

groups of contiguous taxels fail during a physical interaction, causing a set of *blind spots* in the tactile image. In the second case, the analysis is made assuming to eliminate a random distribution of faulty taxels (likewise a salt and pepper noise) from the 2D triangulation, producing a tactile map with lower spatial resolution.

To this aim, two experimental tests have been conducted, simulating: (i) failures of groups of taxels (**Test A**); (ii) randomly distributed faulty taxels (**Test B**). In order to benchmark these experiments we used the models **HandsNet** and **SegNet** for the classification and segmentation task, respectively, which performed best in Section 8.

Furthermore, an additional experiment (**Test C**) has been conducted to analyze how the hand recognition system behaves when applied on sensorized robot parts having a significantly different geometry.

9.1. Test A

The goal of this experiment is to evaluate the performance of the proposed method when groups of contiguous tactile elements stop working, possibly at run time. In this scenario, it is assumed that the response of the faulty taxels is zero producing a sort of *blind spot* in the tactile map. The problem of detecting faulty taxels and to set the corresponding

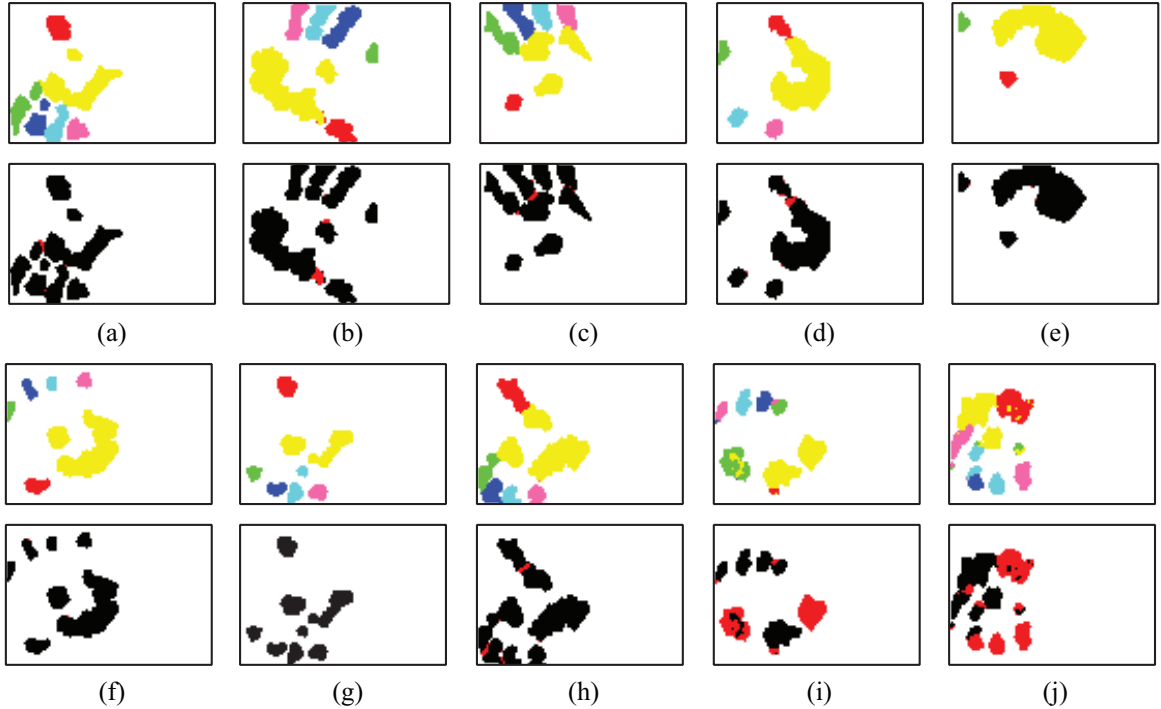


Fig. 13. Examples of segmentation results. mAcc: (a) 98.88%; (b) 94.51%; (c) 98.00%; (d) 94.17%; (e) 100.00%; (f) 100.00%; (g) 100.00%; (h) 96.87%; (i) 72.21%; (j) 41.51%. First line: SegNet output. Second line: thresholded tactile image with red areas corresponding to misclassified pixels.

measurements to zero is part of the data acquisition and the processing pipeline and it is beyond the scope of this article.

Several tactile maps affected by randomly generated patterns of faulty taxels (i.e., *corrupted maps*) have been considered. For each contact, corresponding to images belonging to the test sets described in Section 7, a new tactile image has been regenerated using the corrupted map for both the classification and segmentation tasks. Then, the performance of the models has been evaluated on these new test sets of images. The failure patterns have been created using the following procedure: a taxel lying on the tactile map is randomly selected as the center of the blind spot, then the response of all the taxels within a distance of \bar{r} is set to zero. The number of blind spots N_s corrupting a tactile map can range from 1 to 4, whereas the radius of the spots \bar{r} varies from 10 to 40 mm in steps of 10 mm. For each one of the 16 combinations of these parameters, 10 random patterns have been generated, leading to a total of 160 corrupted maps. Examples of corrupted maps with different values of N_s and \bar{r} are shown in Figure 14. The full list of corrupted tactile maps is included as supplementary material.

In order to evaluate the performance in the case of the segmentation task, the same blind spots appearing on the test images have been transferred to the ground truth images.

Tables 11 and 12 show the performance for each combination of N_s and \bar{r} , computed by averaging the results obtained for the corresponding 10 random patterns. From

Tables 11 and 12 it can be seen that in the classification case the system provides an acceptable performance even with high levels of degradation. In the case of the segmentation task, the proposed method is less robust, providing a mean accuracy of about 80% in the worst case.

9.2. Test B

After a failure is detected and there is no contact occurring, the faulty taxels can be removed from the tactile map and the triangulation can be recomputed, thus generating a tactile map with *lower* spatial resolution. In this experiment, a salt and pepper faulty pattern is simulated, randomly removing from the tactile map a certain percentage \bar{p} of the taxels. The goal is to benchmark the system, evaluating its dependency on the spatial resolution of the tactile map. The percentage of removed taxels \bar{p} is a parameter which varies from 10% to 70% with steps of 10%. Taxels are *incrementally* removed. This means that the taxels lying on the tactile map generated with 20% of faulty sensors are a subset of the ones generated with 10%.

Once the taxels are removed from the tactile map, the triangulation is recomputed. In addition in this case, 10 patterns of broken sensors are randomly generated for each percentage value; therefore, 70 different tactile maps have been created and for each one a corresponding dataset of tactile images has been generated. Figure 15 shows examples of the degradation obtained for different percentage of

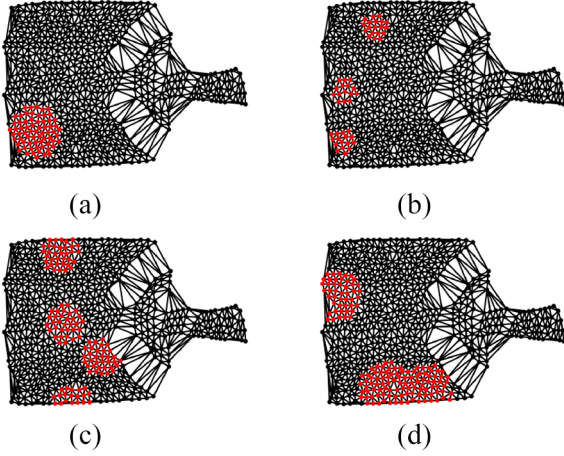


Fig. 14. Examples of corrupted tactile maps: (a) $N_s = 1$ and $\bar{r} = 40$; (b) $N_s = 3$ and $\bar{r} = 20$; (c) $N_s = 4$ and $\bar{r} = 30$; (d) $N_s = 3$ and $\bar{r} = 40$. Red areas corresponds to contiguous regions of faulty taxels.

Table 11. Test A: classification. Mean scores obtained over the 10 test sets for each combination of number of spots and radius values.

N_s	\bar{r} (mm)	Accuracy
1	10	97.81%
1	20	97.50%
1	30	97.61%
1	40	97.21%
2	10	97.68%
2	20	97.28%
2	30	96.88%
2	40	95.18%
3	10	97.73%
3	20	97.00%
3	30	96.16%
3	40	93.95%
4	10	97.67%
4	20	96.78%
4	30	94.42%
4	40	89.74%

removed taxels. The full list of downsampled tactile maps is included as supplementary material.

The benchmark for the segmentation task requires labeled ground truth images (see Section 7.2). As the tactile maps have changed, to exactly evaluate the performance of the segmentation model it would require all 70 of the tactile images in the dataset to be labeled pixel-wise: this is practically an infeasible operation. In order to overcome this issue, for each low-resolution tactile images, the following procedure has been applied. Given \mathbf{I}_H^T the segmented ground truth image at full resolution (see Section 7), and given \mathbf{I}_L^O the corresponding tactile image generated from a low-resolution map, a binary mask is computed as

$$\mathbf{I}_M = [\mathbf{I}_H^T]^B \wedge [\mathbf{I}_L^O]^B$$

Table 12. Test A: segmentation. Mean scores obtained over the 10 test sets for each combination of number of spots and radius values.

N_s	r	Acc	mAcc	mIoU	fwIoU
1	10	93.18%	92.58%	88.66%	90.33%
1	20	92.81%	92.07%	88.03%	89.91%
1	30	92.48%	91.33%	87.36%	89.59%
1	40	91.53%	90.03%	85.89%	88.58%
2	10	93.14%	92.56%	88.63%	90.28%
2	20	92.61%	91.76%	87.72%	89.67%
2	30	91.70%	90.40%	86.04%	88.65%
2	40	90.22%	88.63%	83.96%	87.06%
3	10	93.03%	92.42%	88.44%	90.16%
3	20	91.81%	90.66%	86.05%	88.68%
3	30	90.05%	88.05%	83.43%	86.90%
3	40	86.42%	82.47%	77.51%	83.05%
4	10	92.90%	92.27%	88.31%	90.03%
4	20	92.01%	90.99%	86.81%	89.06%
4	30	88.76%	85.89%	81.03%	85.42%
4	40	84.15%	80.20%	74.40%	80.52%

where $[\cdot]^B$ is the thresholding operator and \wedge is the logical AND operator. Then the actual low-resolution pair $(\mathbf{I}_L, \mathbf{I}_L^T)$ is computed as

$$\begin{aligned} \mathbf{I}_L &= \mathbf{I}_L^O \circ \mathbf{I}_M \\ \mathbf{I}_L^T &= \mathbf{I}_H^T \circ \mathbf{I}_M \end{aligned}$$

where \circ represents the pixel-wise product. Figure 16 graphically describes this process. Clearly, this is an approximation, because some of the pixels are not considered. However, it gives a *qualitative* assessment of the results obtained when lowering the resolution of the tactile map.

Tables 13 and 14 list the accuracy of the models described in Section 8, evaluated on the low-resolution test sets. Similarly to **Test A**, the scores are computed by averaging the results obtained on the 10 datasets generated for each \bar{p} value. In Table 14, the quantity p_d represents the mean percentage of pixels discarded from the low-resolution image as a result of the masking operation described previously.

The results obtained from this experiment show that the system is robust with respect to changes in spatial resolution of the sensors. Indeed, even with 60% of taxels removed, the system provides a classification accuracy above 90%. In the case of the segmentation task, a mean accuracy higher than 90% can be achieved considering 30% of faulty taxels.

9.3. Test C

To test the transferability of the proposed method, a custom end-effector for the Baxter robot has been designed. The new part is shown in Figure 17, along with its tactile map and an example of a tactile image generated from a human hand contact. As it can be seen, the contacts on this tactile map are mapped generating tactile images

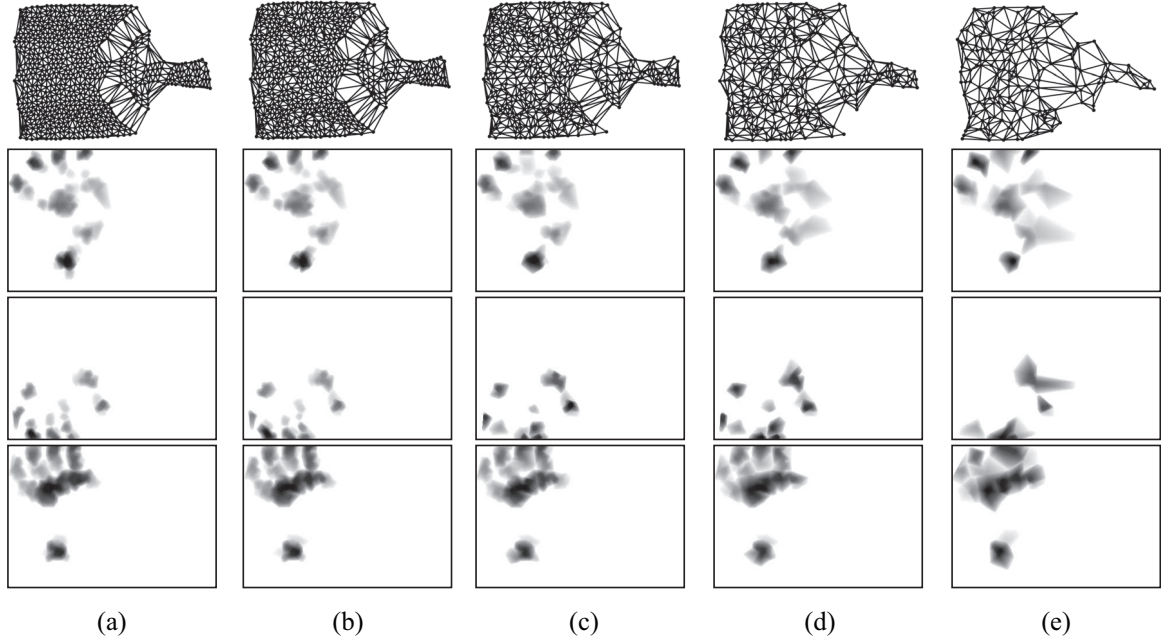


Fig. 15. Examples of downsampled tactile maps and tactile images generated for different values of \bar{p} : (a) original; (b) $\bar{p} = 10\%$; (c) $\bar{p} = 30\%$; (d) $\bar{p} = 50\%$; (e) $\bar{p} = 70\%$. The first row shows the tactile maps, whereas the remaining rows show the level of degradation of the tactile images generated from the corresponding tactile map.

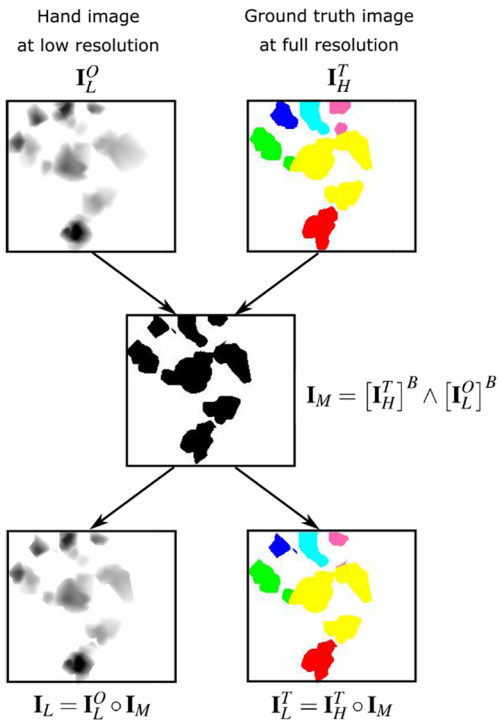


Fig. 16. Process for generating data to evaluate the segmentation model with low-resolution tactile maps. In the example, the hand image is generated from a tactile map where 40% of the taxels have been removed.

completely different from the ones used for training the models in Section 5.

Table 13. Test B: classification Mean scores obtained over the 10 test sets for each value of \bar{p} .

\bar{p}	Accuracy
10%	97.33%
20%	96.98%
30%	96.63%
40%	95.93%
50%	94.79%
60%	92.67%
70%	88.67%

Table 14. Test B: segmentation. Mean scores obtained over the 10 test sets for each value of \bar{p} .

\bar{p}	Acc	mAcc	mIoU	fwIoU	p_d
10%	92.75%	91.69%	87.83%	89.85%	6.05%
20%	92.59%	91.16%	87.31%	89.65%	11.47%
30%	92.17%	90.54%	86.62%	89.21%	16.40%
40%	91.77%	89.55%	85.69%	88.79%	20.88%
50%	90.71%	87.62%	83.48%	87.43%	25.41%
60%	89.13%	84.88%	80.23%	85.54%	30.58%
70%	84.83%	78.20%	72.83%	80.84%	35.87%

Considering the classification task, to validate the **HandsNet** model on this new geometry, a new dataset is required. The end-effector has been attached to the robot and a new dataset has been collected following the same procedure described in Section 7.2. These new experiments

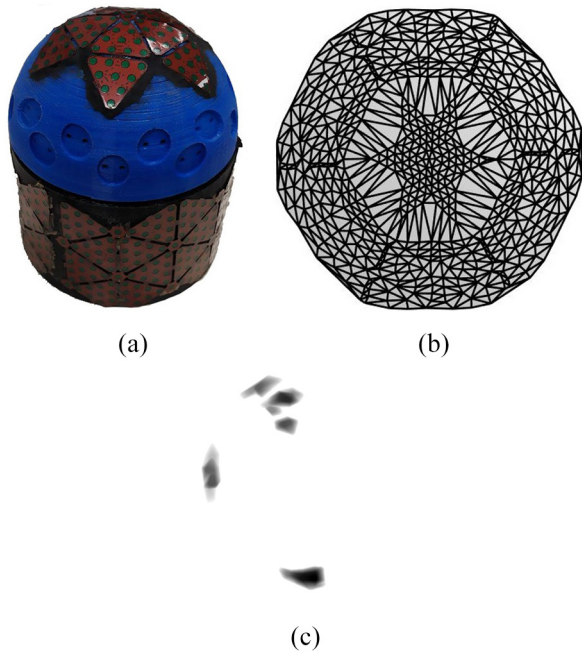


Fig. 17. The sensorized robot end-effector used in this experiment. (a) Robot end-effector partially covered with tactile sensors. (b) The robot end-effector tactile map. (c) Example of a tactile images generated by a human touching the end-effector.

involved 12 people, leading to a new dataset composed of 228 hand images and 250 non-hand images.

A first experiment consisted in feeding the model considering the whole amount of images as a new test set. This produced very poor results, with a mean accuracy below 53%. As it can be seen from Table 15, almost all the hand contacts are misclassified, which is reasonable, since the human hand shape is mapped in a completely different way with respect to the original case.

A possible solution to obtain better results would be to perform fine tuning, allowing the model to learn the newly introduced distortions. Thus, the new dataset has been split into training and test sets using the same modalities described in Section 7.2. Then a fine-tuning of the **HandsNet** model has been performed using the training set. The model has been trained on the new data for 120 epochs using a batch size of 128 and a learning rate of 0.01, which has been halved after 60 epochs. The learning rate applied during the training has been reduced of a 0.1 factor in the first two convolutional layers. The training process led to a mean accuracy higher than 93%. In this phase an intensive hyper-parameters tuning procedure has not been performed. Table 16 shows the confusion matrix of the model fed with the test set.

10. Conclusions

In this work, a technique allowing to discriminate between human hand contacts and other generic type of contacts has

Table 15. Confusion matrix of the **HandsNet** model fed with the images generated from the robot end-effector tactile map. The mean accuracy is 52.92%.

	Hand	Non-hand
Hand	8.71%	2.87%
Non-hand	91.29%	97.13%

Table 16. Confusion matrix of the **HandsNet** model after the fine-tuning procedure. Results are computed on the test set of tactile images generated from contact occurring on the robot end-effector. The mean accuracy is 93.57%.

	Hand	Non-hand
Hand	92.41%	5.26%
Non-hand	7.59%	94.74%

been proposed. Furthermore, it has been shown that human hand contacts can be segmented with a good accuracy to recognize the various hand parts involved into the contact.

With respect to the existing literature, mostly based on the processing of planar tactile measurements, our approach is based on the transformation of tactile pressure measurements obtained from taxels non-uniformly placed on curved robot body parts. This leads to a 2D tactile image which can be processed and classified using state-of-the-art image processing techniques.

The results of this article can have a major impact in the domain of pHRI because the recognition of a human hand contact can be seen as a voluntary interaction aimed at starting a cooperation. Moreover, the possibility of segmenting the pressure distribution can provide relevant information about the role of the various part of the hand involved in the interaction. An example is given in Figure 18, where it can be seen that, after the segmentation operation, the information related to the contact distribution can be extracted for each part of the hand involved in the contact.

Furthermore, the robustness and the transferability of the proposed method have been analyzed, which, to the best of the authors' knowledge, it is a novel contribution with respect to current tactile processing/classification literature.

The models used in the classification tasks have been implemented using Matlab 2018b, with acceptable time performance with respect to the sampling rate of the tactile images. This suggests that an efficient implementation of the models, using optimized libraries, such as Tensorflow (Abadi et al., 2015), can further speed-up the computation.

It can be observed that the proposed approach is not tied to a specific technology. Indeed, in order to create a tactile image, the major requirement is to have a discrete distribution of contact measurements on the robot body.

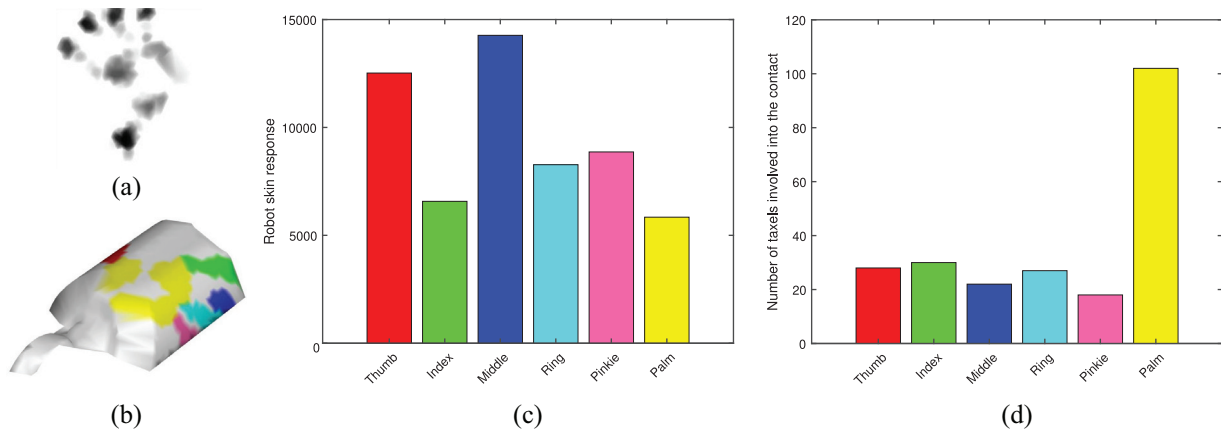


Fig. 18. The segmentation is used for retrieving contact properties for each part of the hand involved in the interaction. (a) Original tactile image. (b) Segmented areas mapped on the 3D model of the robot. (c) Mean pressure distribution for each part of the hand. The scale can range from 0 to 65,536 (see Section 7.1). (d) Number of taxels involved into the contact for each part of the hand.

The results of this article represent the stand point for further research. First by considering the problem of multiple contacts. Second, addressing the problem of recognizing the type of pHRI (e.g., push, pull, twist, etc.) by analyzing the contact dynamics considering sequences of tactile images.

Funding

The research leading to these results has received funding from the European Community's Framework Programme Horizon 2020 (grant agreement number 820767, project CoLLaboratE).


Supplementary material

Datasets, tactile maps, and output of the classification and segmentation models are included as supplementary materials. Further details can be found in the provided ReadMe files.

Note

1. Each subject signed an informed consent form and all the data have been carefully anonymized.

ORCID iD

Alessandro Albini  <https://orcid.org/0000-0003-1562-7044>

References

- Abadi M, Agarwal A, Barham P, et al. (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. Available at: <https://www.tensorflow.org/>.
- Albini A and Cannata G (2018) Tactile images generation from contacts involving adjacent robot links. In: *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 306–312.
- Albini A, Denei S and Cannata G (2017a) Enabling natural human–robot physical interaction using a robotic skin feedback and a prioritized tasks robot control architecture. In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 99–106.

- Albini A, Denei S and Cannata G (2017b) Human hand recognition from robotic skin measurements in human–robot physical interactions. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4348–4353.
- Badrinarayanan V, Kendall A and Cipolla R (2017) Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(12): 2481–2495.
- Beyerer J, Puente León F and Frese C (2016) *Morphological Image Processing*. Berlin: Springer, pp. 607–647.
- Bicchì A, Salisbury JK and Brock DL (1993) Contact sensing from force measurements. *The International Journal of Robotics Research* 12(3): 249–262.
- Billard A, Calinon S, Dillmann R and Schaal S (2008) Robot programming by demonstration. In: *Springer Handbook of Robotics*. Berlin: Springer, pp. 1371–1394.
- Cannata G, Denei S and Mastrogiovanni F (2010) Tactile sensing: Steps to artificial somatosensory maps. In: *19th International Symposium in Robot and Human Interactive Communication*, pp. 576–581.
- Cannata G, Maggiali M, Metta G and Sandini G (2008) An embedded artificial skin for humanoid robots. In: *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 434–438.
- Cao L, Kotagiri R, Sun F, Li H, Huang W and Aye ZMM (2016) Efficient spatio-temporal tactile object recognition with randomized tiling convolutional networks in a hierarchical fusion strategy. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, pp. 3337–3345.
- Cheung E and Lumelsky V (1989) Development of sensitive skin for a 3D robot arm operating in an uncertain environment. In: *Proceedings, 1989 International Conference on Robotics and Automation*, Vol. 2, pp. 1056–1061.
- Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: *CVPR09*.
- Desbrun M, Meyer M and Alliez P (2002) Intrinsic parameterizations of surface meshes. *Computer Graphics Forum* 21(3): 209–218.
- Dhanachandra N, Manglem K and Chanu YJ (2015) Image segmentation using *k*-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science* 54: 764–771.

- Duchaine V and Gosselin CM (2007) General model of human–robot cooperation using a novel velocity based variable impedance control. In: *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pp. 446–451.
- Ebert DM and Henrich DD (2002) Safe human–robot-cooperation: Image-based collision detection for industrial robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 1826–1831.
- Erhan D, Bengio Y, Courville A, Manzagol PA and Vincent P (2010) Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11: 625–660.
- Farfåde SS, Saberian MJ and Li LJ (2015) Multi-view face detection using deep convolutional neural networks. In: *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. New York: ACM Press, pp. 643–650.
- Fortune S (1997) Voronoi diagrams and Delaunay triangulations. In: *Handbook of Discrete and Computational Geometry*. Boca Raton, FL: CRC Press, pp. 377–388.
- Friedman J, Hastie T and Tibshirani R (1998) Additive logistic regression: A statistical view of boosting. *Annals of Statistics* 28: 2000.
- Frigola M, Casals A and Amat J (2006) Human–robot interaction based on a sensitive bumper skin. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 283–287.
- Gandarias JM, Gómez-de Gabriel JM and García-Cerezo AJ (2018) Enhancing perception with tactile object recognition in adaptive grippers for human–robot interaction. *Sensors* 18(3): 692.
- García-García A, Orts-Escolano S, Oprea S, Villena-Martínez V, Martínez-González P and García-Rodríguez J (2018) A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing* 70: 41–65.
- Goodfellow I, Bengio Y and Courville A (2016) *Deep Learning*. Cambridge, MA: MIT Press.
- Grau V, Mewes AUJ, Alcaniz M, Kikinis R and Warfield SK (2004) Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on Medical Imaging* 23(4): 447–458.
- Grunwald G, Schreiber G, Albu-Schaffer A and Hirzinger G (2003) Programming by touch: The different way of human–robot interaction. *IEEE Transactions on Industrial Electronics* 50(4): 659–666.
- Guo Y, Liu Y, Georgiou T and Lew MS (2018) A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval* 7(2): 87–93.
- Guo Y, Liu Y, Oerlemans A, Lao S, Wu S and Lew MS (2016) Deep learning for visual understanding: A review. *Neurocomputing* 187: 27–48.
- Haddadin S, Albu-Schaffer A, Luca AD and Hirzinger G (2008) Collision detection and reaction: A contribution to safe physical human–robot interaction. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3356–3363.
- Kaboli M, Long A and Cheng G (2015) Humanoids learn touch modalities identification via multi-modal robotic skin and robust tactile descriptors. *Advanced Robotics* 29(21): 1411–1425.
- Kato H and Harada T (2014) Image reconstruction from bag-of-visual-words. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kim HJ, Lee JS and Park JH (2008) Dynamic hand gesture recognition using a CNN model with 3D receptive fields. In: *2008 International Conference on Neural Networks and Signal Processing*, pp. 14–19.
- Kimura H, Horiuchi T and Ikeuchi K (1999) Task-model based human robot cooperation using vision. In: *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients*, Vol. 2, pp. 701–706.
- Krizhevsky A, Sutskever I and Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: F Pereira, CJC Burges, L Bottou and KQ Weinberger (eds.) *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., pp. 1097–1105.
- Leboutet Q, Dean-León E and Cheng G (2016) Tactile-based compliance with hierarchical force propagation for omnidirectional mobile manipulators. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 926–931.
- Li Y (2012) Hand gesture recognition using Kinect. In: *2012 IEEE International Conference on Computer Science and Automation Engineering*, pp. 196–199.
- Liang H, Yuan J and Thalmann D (2012) 3D fingertip and palm tracking in depth image sequences. In: *Proceedings of the 20th ACM International Conference on Multimedia (MM '12)*. New York: ACM Press, pp. 785–788.
- Lin HI, Hsu MH and Chen WK (2014) Human hand gesture recognition using a convolution neural network. In: *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1038–1043.
- Liu H, Greco J, Song X, Bimbo J, Seneviratne L and Althoefer K (2012a) Tactile image based contact shape recognition using neural network. In: *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 138–143.
- Liu H, Song X, Nanayakkara T, Seneviratne LD and Althoefer K (2012b) A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor. In: *2012 IEEE International Conference on Robotics and Automation*, pp. 1410–1415.
- Liu H, Yu Y, Sun F and Gu J (2017) Visual–tactile fusion for object recognition. *IEEE Transactions on Automation Science and Engineering* 14(2): 996–1008.
- Long J, Shelhamer E and Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.
- Luo S, Mou W, Althoefer K and Liu H (2015a) Localizing the object contact through matching tactile features with visual map. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3903–3908.
- Luo S, Mou W, Althoefer K and Liu H (2015b) Novel tactile-SIFT descriptor for object shape recognition. *IEEE Sensors Journal* 15(9): 5001–5009.
- Minato T, Yoshikawa Y, Noda T, Ikemoto S, Ishiguro H and Asada M (2007) Cb2: A child robot with biomimetic body for cognitive developmental robotics. In: *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pp. 557–562.

- Mittendorfer P and Cheng G (2011) Humanoid multimodal tactile-sensing modules. *IEEE Transactions on Robotics* 27(3): 401–410.
- Mizuuchi I, Yoshikai T, Sodeyama Y, et al. (2006) Development of musculoskeletal humanoid Kotaro. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006 (ICRA 2006)*, pp. 82–87.
- Morar A, Moldoveanu F and Gröller E (2012) Image segmentation based on active contours without edges. In: *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*, pp. 213–220.
- Mukai T, Onishi M, Odashima T, Hirano S and Luo Z (2008) Development of the tactile sensor system of a human-interactive robot “Ri-Man”. *IEEE Transactions on Robotics* 24(2): 505–512.
- Muscari L, Seminara L, Mastrogianni F, Valle M, Capurro M and Cannata G (2013) Real-time reconstruction of contact shapes for large area robot skin. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 2360–2366.
- Nagi J, Ducatelle F, Caro GAD, et al. (2011) Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 342–347.
- Naya F, Yamato J and Shinozawa K (1999) Recognizing human touching behaviors using a haptic interface for a pet-robot. In: *IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2, pp. 1030–1034.
- Ohmura Y, Kuniyoshi Y and Nagakubo A (2006) Conformable and scalable tactile sensor skin for curved surfaces. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006 (ICRA 2006)*, pp. 1348–1353.
- Park S and Kwak N (2017) Analysis on the dropout effect in convolutional neural networks. In: SH Lai, V Lepetit, K Nishino and Y Sato (eds.) *Computer Vision – ACCV 2016*. Cham: Springer International Publishing, pp. 189–204.
- Pezzementi Z, Plaku E, Reyda C and Hager GD (2011) Tactile-object recognition from appearance information. *IEEE Transactions on Robotics* 27(3): 473–487.
- Raheja JL, Chaudhary A and Singal K (2011) Tracking of fingertips and centers of palm using kinect. In: *2011 Third International Conference on Computational Intelligence, Modelling Simulation*, pp. 248–252.
- Schmidt PA, Maël E and Würtz RP (2006) A sensor for dynamic tactile information with applications in human–robot interaction and object exploration. *Robotics and Autonomous Systems* 54(12): 1005–1014.
- Schmitz A, Maiolino P, Maggiali M, Natale L, Cannata G and Metta G (2011) Methods and technologies for the implementation of large-scale robot tactile sensors. *IEEE Transactions on Robotics* 27(3): 389–400.
- Schneider A, Sturm J, Stachniss C, Reiser M, Burkhardt H and Burgard W (2009) Object identification with tactile sensors using bag-of-features. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 243–248.
- Seminara L, Capurro M and Valle M (2015) Tactile data processing method for the reconstruction of contact force distributions. *Mechatronics* 27: 28–37.
- Silvera-Tawil D, Rye D and Velonaki M (2015) Artificial skin and tactile sensing for socially interactive robots: A review. *Robotics and Autonomous Systems* 63: 230–243.
- Simonyan K and Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.
- Someya T, Sekitani T, Iba S, Kato Y, Kawaguchi H and Sakurai T (2004) A large-area, flexible pressure sensor matrix with organic field-effect transistors for artificial skin applications. *Proceedings of the National Academy of Sciences* 101(27): 9966–9970.
- Stiehl WD and Breazeal C (2005) Affective touch for robotic companions. In: J Tao, T Tan and RW Picard (eds.) *Affective Computing and Intelligent Interaction*. Berlin: Springer, pp. 747–754.
- Sudre CH, Li W, Vercauteren T, Ourselin S and Jorge Cardoso M (2017) Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: MJ Cardoso, T Arbel, G Carneiro, et al (eds.) *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham: Springer, pp. 240–248.
- Tawil DS, Rye D and Velonaki M (2011) Improved image reconstruction for an eit-based sensitive skin with multiple internal electrodes. *IEEE Transactions on Robotics* 27(3): 425–435.
- Tawil DS, Rye D and Velonaki M (2012) Interpretation of the modality of touch on an artificial arm covered with an EIT-based sensitive skin. *The International Journal of Robotics Research* 31(13): 1627–1641.
- Um D, Stankovic B, Giles K, Hammond T and Lumelsky V (1998) A modularized sensitive skin for motion planning in uncertain environments. In: *Proceedings 1998 IEEE International Conference on Robotics and Automation*, Vol 1, pp. 7–12.
- Wasko W, Albini A, Maiolino P, Mastrogianni F and Cannata G (2019) Contact modelling and tactile data processing for robot skins. *Sensors* 19(4): 814.
- Wosch T and Feiten W (2002) Reactive motion control for human–robot tactile interaction. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3807–3812.
- Yang C and Lepora NF (2017) Object exploration using vision and active touch. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6363–6370.
- Yosinski J, Clune J, Bengio Y and Lipson H (2014) How transferable are features in deep neural networks? In: Z Ghahramani, M Welling, C Cortes, ND Lawrence and KQ Weinberger (eds.) *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc., pp. 3320–3328.

Appendix. Training details

This appendix reports information about the training procedure and the selection of hyper-parameters.

The methodology adopted to find a good set of hyper-parameters is the same among the models. In particular, each model has been subjected to a tuning procedure, where the effects of several possible combinations of hyper-parameters have been investigated. Each combination has been evaluated using *five-fold cross-validation* on the training data (Goodfellow et al., 2016). During the process, a *dataset augmentation* is performed on the training folds: the images have been flipped both horizontally and

vertically, increasing the number of images in the training fold by a factor of three.

During the experiments, it was observed that the tactile image size has an effect on the model performance. Thus, we decided to treat it as a hyper-parameter to be tuned.

Images at different resolution have been tested and in the end that with a resolution of 68×100 has been kept because it provided the best scores among the models.

As discussed in Section 3, the shape of the tactile images is 247×362 . In the case of the **HandsNet**, **BoVW**, and **SegNet** we can directly resize and feed images of 68×100 pixels. On the other hand, the **VGG16 + SVM**, **VGG16 + ft**, and **FCN**, require an input of 224×224 . In order to work with inputs having the same resolution, the tactile images of 68×100 pixels have been padded with zeros in order to fit the shape of 224×224 .

A.1. Human hand touch classification

The two networks have been trained in order to minimize the *cross-entropy loss* (Goodfellow et al., 2016), using the hyper-parameters reported in Table 17, where **lr** is the initial learning rate and **lrdf** is a drop factor applied to the learning rate every **lrde** epochs. The other hyper-parameters are the batch size, and the number of training epochs.

For what concerns the **VGG + ft** net, the learning rate, defined by the parameters reported in Table 17, has been applied only in the classification layers. Furthermore, during the training process, the value of the learning rate has been decreased of a 0.1 factor, to fine tune the first three convolutional layers of VGG16.

In the **VGG + SVM** model, the network works as a feature extractor, so there is no need for training. The classification is performed using a linear SVM, which has been selected by tuning the penalty parameter C . The classifier with $C = 0.25$ has been selected, because it provided the highest accuracy.

In the case of BoVW model, the hyper-parameters considered are length of the SIFT descriptors L (Lowe, 2004) and the vocabulary size K (Kato and Harada, 2014), which have been selected as 128 and 80, respectively.

A.2. Human hand touch segmentation

As can be seen in Figure 10 the class distribution is not uniform, indeed most of the pixels (almost 40%) are labeled as

Table 17. Hyper-parameters used to train the networks for the classification task.

Model	lr	lrdf	lrde	Batch size	Epochs
HandsNet	0.01	0.2	40	64	80
VGG + ft	0.1	0.5	40	32	80

Table 18. Hyper-parameters used to train the networks for the semantic segmentation task.

Model	lr	lrdf	lrde	Batch size	Epochs
SegNet	0.1	0.1	90	16	100
FCN	0.1	0.15	80	8	130

Palm. A non-balanced dataset can cause problems during the training phase because the learning process can be biased in favor of the *Palm* class. As suggested in the literature (Badrinarayanan et al., 2017; Sudre et al., 2017) there are two efficient strategies to deal with an imbalanced dataset. One solution is to use a *cross-entropy* loss weighted using the *median frequency balancing*. Another approach is to use the *dice* loss function. Both methods have been tested. In the case of **SegNet**, the weighted cross-entropy loss performed better, thus it has been selected for training the model. In contrast, the dice loss produced better results with the **FCN** model.

As described in Long et al. (2015) there are three versions of the **FCN**, namely FCN-32s, FCN-16s, and FCN-8s. The difference among them is the size of the stride used in the classification layer. According to Long et al. (2015) the 8s version provides slightly more accurate predictions. In this work, we trained the FCN-16 because with our data we did not find any improvement with respect to use FCN-8s, which has a higher computational cost.

The hyper-parameters selected after the tuning procedure are reported in Table 18. During the training, the models have been fine-tuned by reducing the applied learning rate of a 0.1 factor in the VGG16 convolutional layers, slightly adapting their weights to the new data.