Object recognition using tactile sensing in a robotic gripper

Vladimir Riffo[1*], Christian Pieringer, Sebastián Flores[1] and Carlos Carrasco[1]

[1]DIICC, Universidad de Atacama, Av. Copayapu 485, Copiapó. Atacama, Chile.

*Corresponding author. E-mail(s): vladimir.riffo.b@gmail.com; Contributing authors: christian.pieringer@gmail.com; sebastian.flores@uda.cl; carlos.carrasco@alumnos.uda.cl

Abstract

Object recognition using the tactile sense is one of the leading human capacities. This capability is not as developed in robotics as other sensory abilities, *e.g.*, visual recognition. In addition to the robot's ability to grasp objects without damaging them, it is also helpful to provide these machines with the ability to recognize objects while gently manipulating them, as humans do in the absence of or complementary to other senses. Advances in sensory technology have allowed us to detect different types of environments accurately. However, the challenge of being able to represent sensory information efficiently persists. We propose a sensory system that allows a robotic gripper armed with pressure sensors to recognize objects through tactile manipulation. We design a pressure descriptor to characterize the voltage magnitudes across different objects and, finally, use machine learning algorithms to recognize each object category. The results show that our pressure descriptor characterizes the different categories of objects in this experimental setup. Our system can complement other sensory data to perform different tasks in a robotic environment and we propose future research areas to handle tactile manipulation problems.

Keywords: tactile sensing, object recognition, machine learning, non-destructive testing

1 Introduction

Humans have a complex multi-sensory system that allows them to interact with the surrounding unstructured environment and perform complicated tasks. Currently, several production processes still require human manipulation. In some cases, these processes need palpations for quality assurance detecting irregularities or damages, as in the classification of fruit maturity states. Furthermore, other processes are dangerous and involve risk to life, *e.g.* the handling of radioactive materials [1]. In general, human performance in inspection tasks decreases because of stress, fatigue or illness, among other factors [2]. This fact illustrates the point that robotic manipulation that allows for recognizing and sensing the manipulated object can improve automation. For these reasons, it is necessary to improve the sensitivity of robotic grippers while providing security to human inspectors and enhancing the processes' automation. We believe that progress in this research area would be helpful to expand the usages of Non-Destructive Testing (NDT) to processes still strongly performed by humans.

Recent advances in robotics demonstrate the potential uses of robots or autonomous agents in many areas of our lives, for instance self-driving vehicles [3–5], medical applications [6, 7], social robotics [8], and autonomous underwater vehicle (AUV) systems [9, 10], among others. In this sense, robots have achieved consistent progress in mimicking human abilities of sensing and decision making, *i.e.* the robot's sensors can acquire stimuli from the environment and decide as people do through their smell, vision and skin. The improvements and development of sensing technologies have helped robots become more precise in sensing various stimuli perceived by humans, *e.g.* in gas detectors [11], light sensors [12] and pressure [13].

This work describes a methodology for recognizing objects using a robotic manipulator armed with a set of strain gauges arranged in a Wheatstone bridge configuration as pressure sensors. Each sensor provides a voltage signal according to the surface of the objects. We collect a series of voltage measurements during data acquisition while the gripper puts pressure on the test object. Then, in our process pipeline, the system transforms the voltage levels into a feature descriptor to characterize each object's category. We design this descriptor to represent the voltage variations. We use this data as input for a set of classification algorithms: *i*) Decision Tree, *ii*) Naive Bayes (NB), *iii*) Neural Networks (NN), *iv*) *k*-Nearest Neighbours (*k*-NN) and *v*) Support Vector Machine (SVM). We implement all these algorithms using the Python module Scikit-learn [14]. Our method allows for object recognition with different types of harness. Along with this research, we address two problems: *i*) recognize objects using pressure sensors, emulating human execution, and *ii*) automate inspection processes or eventually provide an analytic tool for people working in production lines with manual inspection providing quality indices, such as fruit exporting companies.

A literature review on tactile manipulation and recognition identifies three main research areas: centred on the construction and performance of grippers

[15, 16]; focused on pressure sensors [17, 18]; and finally, focused on object recognition through grippers [13]. In the latter research, the use of the *k*-NN algorithm provides the best results. In this study, we extend the work in [13], incorporating another type of sensor and another set of machine learning algorithms. The results show that these additional changes improve the performance in recognition.

We carry out experiments with a limited database (similar to the database used in [13]). We manually collect and assign labels to ten object categories: computer mouse, glue container, rubber duck, smartphone, bell pepper, ball of wool, stress ball, dishwashing sponge, can of soda and apple. All of these objects present a wide variability, varying from rigid to flexible. We use the True Positive Rate (TPR) and the False Positive Rate (FPR) performance indices to quantify and evaluate our methodology's best settings. Our results show the best performance in recognition of two categories using the SVM classifier: soda cans, with a TPR = 100% and FPR = 0%, and in apples, with a TPR = 100% and FPR = 3%. These results are encouraging and support the use of our method in environments that require handling processes.

This article presents the following three contributions:

- A low-cost method for data acquisition using tactile sensors,
- A tactile descriptor for object recognition in the context of NDT,
- A database of objects with variable surface properties.[1]

The document is organized as follows. Section 2 presents related work on tactile sensors and concepts involved in our work. Section 3 introduces and describes our approach theoretically and summarizes the classification method used in the experimental test. Section 4 describes the experimental setup and the results of using our method. Finally, Section 5 outlines our conclusions, main remarks and further improvements.

2 Related Work

Non-destructive testing is crucial in a production environment that secures the quality of products delivered to final customers [19]. In certain situations, these tests involve the manipulation of objects by human inspectors. Currently, more and more companies rely on automation in the inspection process. Robotic manipulators are preferred mechanisms to mimic the human role during the inspection and manipulation [20–23]. However, robotic manipulators generally lack sensory feedback that lets them know or register information about the manipulated object.

Tactile sensing uses a set of sensors to measure one or more tactile properties through physical touch, such as temperature, shape and texture. These sensors allow the loop to be closed between the manipulator and the decision system. In the last forty years, research on tactile sensing has demonstrated its

[1] See https://www.dropbox.com/s/977e52z0wtyom4o/Data_Base_Tactile_Sensing.rar?dl=0

potential in biomedicine [24, 25], minimally invasive surgery [26, 27] and robotics and manufacturing [23, 28], among others.

Research into tactile sensors involves the use and development of different technologies. Nevertheless, there is a significant cost-effective trade-off to ensure that these technologies enter at industry level [29, 30]. Strain gauges are a low-cost alternative for sensing surfaces. These types of sensors present a particular behaviour of hysteresis that introduces noise during the measuring. A Wheatstone bridge configuration provides immunity to noise to the system [31].

In robotic environments, tactile recognition provides a complementary method to visual recognition to cope with ambiguities. In the last ten years, tactile sensors for material and object recognition have gained attention in classification and inspection tasks. Machine Learning algorithms are a crucial part of recognition systems that contribute as a critical stage for the automatic inspection context [30].

Object recognition through tactile sensors is still an open problem and presents challenging questions about the sensors, inspection sequence and movements, and processing techniques. In 2006, Mazid et al. introduced an opto-tactile sensor able to assess the surface texture of objects as an input to object recognition stages [44]. The authors proposed a mathematical model to relate geometrical parameters to the output voltage. Their results demonstrate its simple construction and viability for integrating this technology on a robotic platform. However, the article did not dive into using the sensor output to ingest recognition stages of classification models. By contrast, our work focuses on the recognition stages, leaving the methodology open to integrating any sensory device to acquire the tactile data. In [13], the authors proposed a robust tactile sensor based on piezoresistive materials and conducting thread electrodes. Materials gave the sensor high repeatability. During an active exploration procedure, a robotic gripper performed palpations on the objects and acquired information to describe and classify the objects. The authors use a small dataset of 12 object classes. The results showed successful sensor measurements in a haptic-based object-classification scenario compared with a well-known industrial sensor. Finally, a KNN model discriminated between the time series of the different pressure patterns. In our work, we use a similar setup as in [13] but extend the results by evaluating a pressure descriptor in a set of machine learning methods. The pressure descriptor is a combination of voltage levels at different gripper rotations. The authors in [33] presented a systematic comparison of features and classification algorithms to recognize 49 real-word objects using data from a BioTac tactile sensor. The analysis included the quantitative comparison of performance using different sets of features such as pressure, statistical moments, temperature, physical movements and temporal features.

Later, the authors in [42] proposed a magnetostrictive tactile sensing system that used an Extreme Learning Machine (ELM). The authors conducted experiments using four different kinds of stiffness, repeating the grasp 30 times for each material. Experiments showed precision and consistency in the output of the magnetic probe. The results also showed the advantages of the ELM model with a low number of training samples. However, the tests were still limited in samples, materials and forms. Our work expands the way to represent the object, including different palpation points.

Recently in [43], Li et al. proposed a quadruple skin-inspire tactile sensor system used in the task of garbage identification. The system integrated four different sensors into a robotic hand: pressure sensing, material thermal conductivity sensing, and bimodal temperature sensing for the robot hand to recognize objects precisely. The system also included a stage of data integration through machine learning techniques. In our work, we focus on the way that we can represent the pressure vector. Instead of using multiple sensors, we include multiple rotations of the robotic hand. We also use a small dataset of objects based on previous work in order to present a prototype of this framework.

Similarly, we use machine learning methods to discriminate between objects and demonstrate the effectiveness of the descriptor as in [13, 33, 42, 43]. We refer the reader to [23, 30, 32, 45] for a detailed review of tactile sensing in robotics and other related areas.

3 Method

This section describes our proposed methodology, including the pressure descriptor based on the tactile sensor measurements, as follows: *i*) Overview, *ii*) Pressure descriptors, *iii*) Characterization and training, *iv*) Classification algorithms, and *v*) Receiver operating characteristic.

3.1 Overview

We frame our research in the context of the automatic inspection of objects. In this sense, our methodology comprises a processing pipeline composed of three stages: data acquisition, data transformation and feature engineering, and classification (Figure 1).

The Data Acquisition stage comprises a robotic gripper armed with pressure sensors that provide an electrical measurement from the palpation.

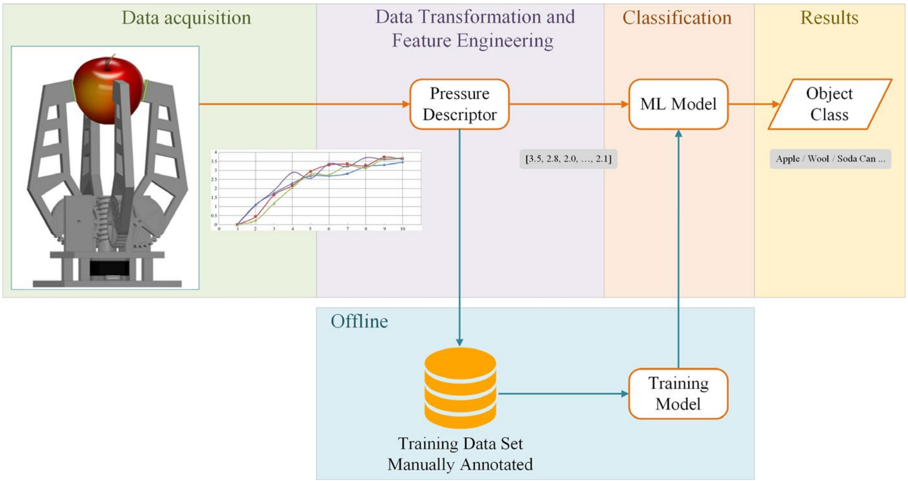

Fig. 1 Diagram of the processing pipeline. It has three stages that range from data acquisition to classification. Training of the models was performed offline and used a manually collected dataset.

The Data Transformation and Feature Engineering stage controls the preprocessing of the electrical signals collected by the gripper and transforms them into a pressure descriptor. Finally, the Classification stage evaluates the descriptor and provides the object's class according to a Machine Learning model previously trained using a training set. Using this pipeline, we manually collected and built a dataset with pressure measurements from ten object classes. The gripper acquired 40 different pressure points during this process by rotating the object for each sample (see Figure 2).

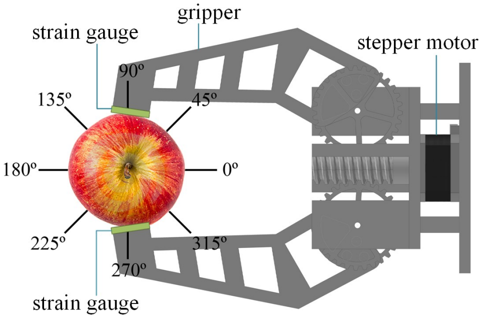

Fig. 2 Diagram of the process of characterization of objects.

3.2 Pressure Descriptor

The acquisition stage yielded a record of the voltage levels from pressure sensors as the object rotated. We proposed a pressure descriptor that organized the samples as a two-dimensional data array, increasing the discriminative capacity of the collected data.

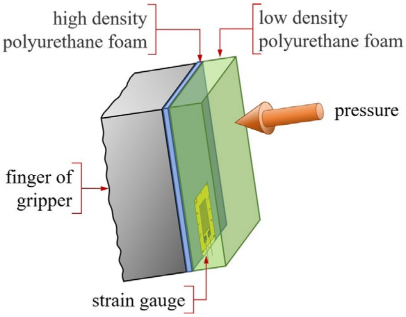

Fig. 3 Detail of one of the gripper's fingers.

The stepper motor performed micro-steps to obtain the pressuredescriptor, causing the gripper fingers to close slowly. Once the gripper made the first contact with the test object, low-density and high-density polyurethane foams began to deform and exert pressure on the strain gauge, as shown in Figure 3. The stepper motor made ten steps, counting from when the first contact with the test object was established, *i.e.* when the system acquired the first voltage level. In this way, the pressure descriptor began to be built (see equation 1).

During acquisition, there was no direct contact between the fingers of the gripper and the object. The motor only rotated in micro-steps, and the low-density and high-density polyurethane foams absorbed the pressure. Therefore, the test object was never damaged. Each pressure level was acquired and set as part of the pressure descriptor.

The descriptor simultaneously stored the pressure values of each object and rotation. The pressure descriptor is defined as F_{ct}, as shown in the equation (1):

$$F_{ct} = \left[f_j^{(i)}\right] = \begin{bmatrix} f_1^{(1)} & f_2^{(1)} & \cdots & f_n^{(1)} \\ f_1^{(2)} & f_2^{(2)} & \cdots & f_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ f_1^{(m)} & f_2^{(m)} & \cdots & f_n^{(m)} \end{bmatrix} \tag{1}$$

where c corresponds to the class of the object varying from $c = 1, \ldots, M$; t corresponds to the test object of class c, which varies from $t = 1, \ldots, N$; i corresponds to the relative position of the measured object, which ranges from $i = 1, \ldots, m$; and j corresponds to the number of pressures exerted on the object, which ranges from $j = 1, \ldots, n$.

In the process of acquiring these measurements, each $f_j^{(i)}$ corresponds to an instantaneous voltage value, which varied according to the pressure exerted on an object of class c and in the position i.

3.3 Characterization and training

We manually placed each object on the gripper during data acquisition until the sensing process finished, smoothly increasing the pressure level j, similarly to how a human would behave. This procedure occurred at four different positions i. For example, as seen in Figure 2, in the case of the apple, the gripper got the first measurements at angles of 0° and 180°, the second at 45° and 225°, the third at 90° and 270° and the last ones at 135° and 315°. For objects with non-spherical or irregular shapes, such as the Smartphone, we removed two of the four gripper fingers, leaving only the gripper fingers with the pressure sensors. Thus, the palpations occurred at different non-coincident points. Finally, for each pair of angles at $j = 1 \ldots 10$, we collected the respective pressure values, as shown in Table 1 and Figure 2. We repeated this procedure for each object, collecting the pressure levels and annotating the object label.

3.4 Classification Algorithms

The dataset formation and model training were an "offline" procedure. We implemented a set of Machine Learning algorithms used for classification: Decision Tree [34], Naïve Bayes (NB) [35], Neural Networks (NNs) [36], *k*-Nearest Neighbours (*k*-NN), and Support Vector Machine (SVM) [37]. We used these five algorithms to evaluate our pressure descriptor's discriminative power and find the classifier that performed best in our processing pipeline for the different object flexibility and stiffness states. In the following subsections, we briefly describe the classifiers included in this study.

3.4.1 Decision tree

Decision trees are information-based models that build a hierarchical structure to provide instance classification [34]. The hierarchical structure defines a set of rules that works as a decision process. The tree begins at a node called the *root* that contains the first attribute that will lead to new rules called *decision nodes*, forming branches until a *terminal node or leaf node* is reached. In this structure,

- A leaf indicates a class,
- A decision node specifies a test to be performed on a single attribute value, with a branch and subtree for each possible outcome.

The traverse of the tree structure eventually (and inevitably) leads to a leaf node that indicates the class to which the instance belongs. Along with the training, Decision Tree algorithms rely on heuristics in order to simplify the tree's structure while keeping the interpretability without compromising accuracy [34], *e.g.* reduced error pruning (REP), pessimistic error pruning (PEP) and minimum error pruning (MEP) [38].

3.4.2 Naive Bayes

Naive Bayes is a probabilistic-based model that relies on Bayes' Theorem and the conditional independence assumption for modelling the conditional probability of a class given the attributes. A *Naive Bayes* classification scheme requires the estimation of the probability density function (pdf) at a point $x = [x(1), \cdots, x(l)]^{\mathsf{T}} \in \mathbb{R}^{l}$, given by:

$$p(x) = \prod_{j=1}^{l} p(x(j)) \tag{2}$$

The probability computation assumes the feature vector components x_j are *statistically independent*. This assumption is helpful in high-dimensional spaces, where many training points must be available to obtain a reliable estimate of the corresponding multidimensional probabilitydensity function. Despite the naive assumption of the feature independence given the class, the overall performance is reasonable since reliable estimates of the 1-dimensional probability density function are still achievable with relatively few data values [39].

3.4.3 Artificial Neural Network

An Artificial Neural Network (ANN) is an error-based model, where the neuron is the fundamental element that tries to imitate the human brain's function. A simple ANN model is the *perceptron*. Each perceptron is excited by a weighted sum over the input signals $x(1), x(2), \ldots, x(l)$ with their corresponding weights $w_1, w_2, \cdots, w_l$, known as synaptic weights by analogy with the terminology used in neuroscience. The weighted sum then passes through an activation function that decides if the neuron "fires", *i.e.* it gives an output value; otherwise, it remains inactive.

The training of an ANN aims to estimate the weights and bias values of all the neuron layers involved in the network based on minimizing a cost function. The most widely used option is the least-squares loss function:

$$J = \sum_{j=1}^{N} \left(y_j - \hat{y}_j\right)^2 \tag{3}$$

where N is the total number of data points in the dataset, y_j is the true class label of $x_j \in \mathbb{R}^l$ and $\hat{y}_j$ is the output of a neural network. In general, the cost function J has several local minima. Therefore, the choice of the initial parameters influences the solution during minimization. In practice, the algorithm is performed several times, starting from different initial values. The weights corresponding to the best solution are the network parameters [36, 39].

3.4.4 *k*-Nearest Neighbours

The *k*-Nearest Neighbours (*k*-NN) is a similarity-based method, where the objective is to estimate the unknown value of the given probability density function at a point x. During the neighbour estimation, the *k*-Nearest Neighbours algorithm performs the following steps:

a) Choose a value for k.
b) Find the distance between x and all training points x_i, $i = 1, 2, \dots, N$. Any measure of distance can be used (*e.g.*, Euclidean, Mahalanobis).
c) Find the k nearest points to x.
d) Compute the volume $V(x)$ in which the k nearest neighbours lie.
e) Compute the estimate by

$$p(x) \approx \frac{k}{NV(x)} \tag{4}$$

If the *k*-NN algorithm uses Euclidean distance and the distance between the furthest neighbour k and x is ρ, the volume $V(x)$ is equal to:

$$V(x) = 2\rho \quad \text{in 1-dimensional space,} \tag{5}$$

$$V(x) = \pi\rho^2 \quad \text{in 2-dimensional space, or} \tag{6}$$

$$V(x) = \frac{4}{3}\pi\rho^3 \quad \text{in 3-dimensional space.} \tag{7}$$

For the more general case of l dimensions and/or the Mahalanobis distance, see [37, 39].

3.4.5 Support Vector Machine

The *Support Vector Machine* (SVM) is another error-based model that aims to solve a non-linear classification task mapping the feature vectors into a larger-dimensional space. We expect the classes to be linearly separable. This mapping is given by:

$$x \mapsto \phi(x) \in H \tag{8}$$

where $x \in \mathbb{R}^l$ and the dimensionality of H is greater than $\mathbb{R}^l$ depending on the choice of (non-linear) $\phi(\cdot)$. Also, if we carefully choose the mapping function from a known family of functions with desirable and specific properties, the inner product between the images $(\phi(x_1), \phi(x_2))$ of two points $x_1$and x_2 can be written as:

$$< \phi(x_1), \phi(x_2) > = k(x_1, x_2) \tag{9}$$

where, $< \cdot, \cdot >$ denotes the operation of the inner product on H, and $k(\cdot, \cdot)$ is a function known as a *kernel* function. Thus, the inner products in the high-dimensional space perform the *kernel function* k over the original low-dimensional space. The space E associated with $k(\cdot, \cdot)$ is known as a Reproducing Kernel Hilbert Space (*RKHS*).

A notable feature of SVM optimization is that all operations are inner products. Therefore, linear problems in high-dimensional space (after mapping) only require the inner products using the corresponding evaluating kernel.

4 Experimental Results

This section describes and shows the experimental setup we used to conduct the experiments to validate our methodology. This setup is a prototype for this research, and future work in this area can include improvements to the setup.

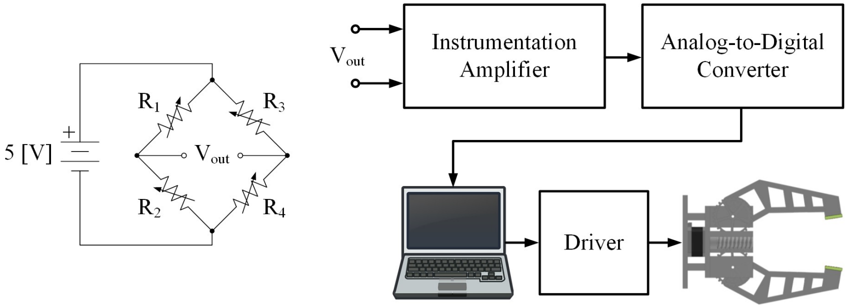

Fig. 4 Data acquisition system.

4.1 Experimental Setup

We developed an acquisition system to evaluate our methodology (see Figure 4). This pipeline starts with a 3D-printed gripper designed by us, motorized by a 5 Volt 1.8-degree stepper motor, model KHP-11m10, manufactured by OKI Electric. This motor has high torque and tiny steps, which favours the palpation of objects, *i.e.* as smoothly as a human hand can. A 5 Volt 1000mA power supply energizes a Toshiba ULN2803APG that drives and protects the stepper motor's coils. In a Wheatstone bridge configuration, the gripper has attached a set of 4 strain gauges of 3.5mm and 120 ohms as a pressure sensor. Two of the four strain gauges R_1 and R_2 are attached to two gripper fingers (opposite fingers). The other two strain gauges, R_3 and R_4, are attached to the static surface of the gripper in such a way that the bridge is in balance, and it will only become unbalanced when R_1 and R_2 deform. A Texas Instruments INA128P instrumentation amplifier acquires the voltage levels from the strain gauge. Finally, an Arduino Uno board acquires and sends the voltage to a computer. Figure 5 shows the complete acquisition system.

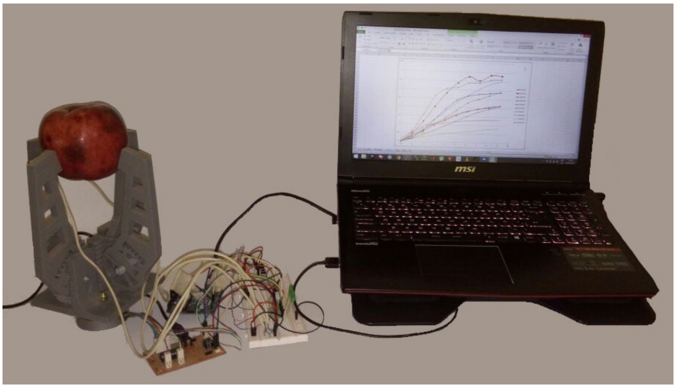

Fig. 5 System used for object inspection.

4.2 Dataset Description

We built a dataset manually by collecting and annotating measurements for ten object classes present in everyday life using our acquisition system. These classes were: computer mouse, glue container, rubber duck, smartphone, bell pepper, ball of wool, stress ball, dishwashing sponge, can of soda, andapple, as shown in Figure 6. The maximum range of the gripper opening constrains the object's size that the system can recognize. In our case, the dataset included objects smaller than 10 centimetres. However, this limitation will depend on the context of the system, *i.e.*, bigger grippers allow for bigger objects.

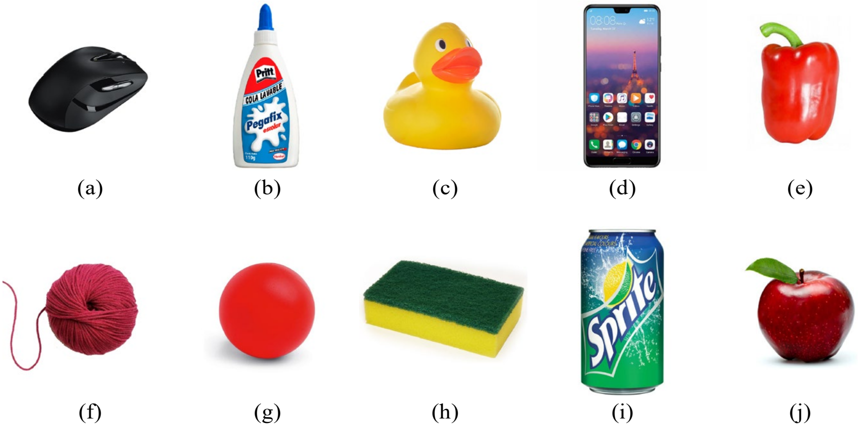

Fig. 6 Representative objects of each of the classes used in this work: (a) computer mouse, (b) glue container, (c) rubber duck, (d) smartphone, (e) bell pepper, (f) ball of wool, (g) stress ball, (h) dishwashing sponge, (i) can of soda, and (j) apple.

The gripper palpated the objects in different positions around an imaginary line in its centre during the acquisition, as shown in Figure 2. For instance, the gripper palpated an apple obtaining four measurements: the first at angles of 0° and 180°, the second at 45° and 225°, the third at 90° and 270°, and the fourth at 135° and 315°. For each pair of angles, the gripper provided ten levels of pressure. Table 1 contains an example of the ten values provided for each rotation of an apple. Figure 7 visualizes the data in Table 1 to show how we created our pressure descriptor graphically. Not all the objects in our research had a spherical or cylindrical shape. In such cases, such as a Smartphone, we proceeded to palpate the 4 points randomly, taking care that none of these points coincided.

Table 1 Values obtained for the descriptor of an apple, that is, F_{ct}, with class $c = 10$ and with the apple object $t = 3$.

Angles (i)	Level of pressure exerted (j)									
	1	2	3	4	5	6	7	8	9	10
0° – 180°	0.020	1.104	1.733	2.290	2.705	2.705	2.827	3.223	3.315	3.452
45° – 225°	0.005	0.449	1.636	2.163	2.944	3.311	3.350	3.276	3.740	3.638
90° – 270°	0.005	0.234	1.182	2.056	2.788	2.769	3.262	3.149	3.647	3.687
135° – 315°	0.005	1.074	1.890	2.881	2.568	3.369	3.223	3.716	3.608	3.691

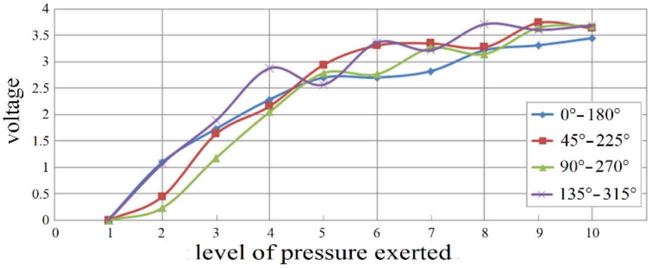

Fig. 7 Graphical representation of the descriptor obtained from the pressure exerted on an apple.

The pressure descriptor allowed us to characterize the different objects used in this research (see Figure 6). Using the acquired measurements and following the representation of equation (1) we have:

$$F_{ct} = [f_j^{(i)}] = \begin{bmatrix} 0.020 & 1.104 & 1.733 & 2.290 & 2.705 & 2.705 & 2.827 & 3.223 & 3.315 & 3.452 \\ 0.005 & 0.449 & 1.636 & 2.163 & 2.944 & 3.311 & 3.350 & 3.276 & 3.740 & 3.638 \\ 0.005 & 0.234 & 1.182 & 2.056 & 2.788 & 2.769 & 3.262 & 3.149 & 3.647 & 3.687 \\ 0.005 & 1.074 & 1.890 & 2.881 & 2.568 & 3.369 & 3.223 & 3.716 & 3.608 & 3.691 \end{bmatrix}$$

Using our experimental setup, we collected ten object samples ($t = 1,\ldots,10$) per class ($c = 1,\ldots,10$) shown in Figure 6. The acquisition system provided the ten levels of pressure exerted ($j = 1,\ldots,10$) per object instance at the four measurements positions ($i = 1,\ldots,4$). Therefore, our training-testing database had a total number of 400 object instances.

4.3 Evaluation and Results

We conducted all the experiments using Python for both of our pipeline's main tasks: data acquisition and data classification. We implemented all the classification algorithms using the Python module Scikit-Learn [14].

Evaluation of the classification algorithm considered the standard performance indicators True Positive Rate (TPR) and Specificity or False Positive Rate (FPR), both widely used in the Machine Learning literature [40] and defined as follows:

$$\mathrm{TPR} = \frac{TP}{N_p} = \frac{TP}{TP + FN} \tag{10}$$

$$\mathrm{FPR} = \frac{FP}{N_n} = \frac{FP}{FP + FN} \tag{11}$$

where TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives, N_p is the total number of positives in the test database, and N_n is the total number of negatives in the test database. TRP and FPR are related to each other in the Receiver Operating Characteristic (ROC) context, where both indicators locate in a two-dimensional coordinate system. The resulting curve will always be monotonic, increasing in both axes. Therefore, the closer the curve is to the upper left corner (TPR = 1 and FPR = 0), the better the detection algorithm. The area under the ROC curve (AUC) summarizes the behaviour of the TPR and FPR [41]. In order to take advantage of the complete dataset, we ran our experiments applying the cross-validation technique using ten folds (90% of the dataset used for training and 10% used for testing). We refer the reader to [40] for more details about model evaluation techniques.

4.3.1 Decision Tree (DT)

Confusion matrix analysis (see Table 2) showed that Decision Trees achieved an average accuracy of 88.25%. The class Can achieved the best classification, obtaining 97.5% of correct classifications. By contrast, the classes mouse, school glue and rubber duck gave the lowest performance (80%).

Table 2 Confusion matrix for the Decision Tree algorithm. The average accuracy for this algorithm was 88.25%.

a	b	c	d	e	f	g	h	i	j	Classified as
80	0	5	2.5	0	0	7.5	0	0	5	a = Mouse
10	**80**	0	0	0	0	0	0	0	10	b = Scholar glue
7.5	0	**80**	0	12.5	0	0	0	0	0	c = Rubber duck
0	0	0	**92.5**	7.5	0	0	0	0	0	d = Smartphone
0	0	5	2.5	**90**	2.5	0	0	0	0	e = Bell pepper
0	0	0	0	0	**92.5**	0	5	2.5	0	f = Wool
2.5	0	2.5	0	0	0	**95**	0	0	0	g = Ball
0	0	0	2.5	0	5	0	**92.5**	0	0	h = Sponge
0	0	0	0	0	2.5	0	0	**97.5**	0	i = Can
7.5	7.5	0	0	0	0	2.5	0	0	**82.5**	j = Apple

4.3.2 Naive Bayes (NB)

For the Naive Bayes classifier, the confusion matrix results indicate that the algorithm achieved an average accuracy of 93.25% (Table 3). This performance is higher than Decision Trees by 5%. Specifically, the Can and Apple classes achieved the best performance (100%) while the classes rubber duck and bell pepper gave the poorest performance, achieving 85% of correct classifications.

Table 3 Confusion matrix for the Naive Bayes algorithm. The Apple class achieved the best performance.

a	b	c	d	e	f	g	h	i	j	Classified as
87.5	0	5	0	0	0	2.5	0	0	5	a = Mouse
0	**97.5**	0	0	0	0	2.5	0	0	0	b = Scholar glue
7.5	0	**85**	0	7.5	0	0	0	0	0	c = Rubber duck
0	0	0	**92.5**	5	0	0	2.5	0	0	d = Smartphone
0	0	5	7.5	**85**	2.5	0	0	0	0	e = Bell pepper
0	0	0	0	2.5	**97.5**	0	0	0	0	f = Wool
2.5	0	2.5	0	0	0	**95**	0	0	0	g = Ball
0	0	0	2.5	0	5	0	**92.5**	0	0	h = Sponge
0	0	0	0	0	0	0	0	**100**	0	i = Can
0	0	0	0	0	0	0	0	0	**100**	j = Apple

4.3.3 Neural Network (NN)

The analysis carried out on the confusion matrix showed that NN correctly classified an average of 93.5% of the instances, being slightly higher than Naive Bayes by 0.25%. The class Can achieved the best performance (100%). The lowest performance for this model occurred for bell pepper, which only achieved 82.5% of instances correctly classified.

Table 4 Confusion matrix for the Neural Network algorithm. The class Can achieved the best performance.

a	b	c	d	e	f	g	h	i	j	Classified as
92.5	0	2.5	0	0	0	2.5	0	0	2.5	a = Mouse
0	**95**	0	0	0	0	2.5	0	0	2.5	b = Scholar glue
2.5	0	**97.5**	0	0	0	0	0	0	0	c = Rubber duck
0	0	0	**90**	5	0	0	5	0	0	d = Smartphone
0	0	5	10	**82.5**	0	0	0	2.5	0	e = Bell pepper
0	0	0	0	0	**97.5**	0	2.5	0	0	f = Wool
0	2.5	5	0	0	0	**92.5**	0	0	0	g = Ball
0	0	0	2.5	0	2.5	0	**95**	0	0	h = Sponge
0	0	0	0	0	0	0	0	**100**	0	i = Can
2.5	5	0	0	0	0	0	0	0	**92.5**	j = Apple

4.3.4 *k*-Nearest Neighbours (k-NN)

The analysis carried out on the confusion matrix (see Table 5) showed that KNN achieved an average of 94.25% (377 hits) of instances correctly classified, being superior to NN by 0.75% and superior to DT by 6%. The classes Rubber duck and Can achieved the best performance (100%). The lowest classification was with bell pepper, reaching 85% of correct classifications.

Table 5 Confusion matrix for the *k*-NN algorithm. The model achieved the best performance for the classes Rubber Duck and Can.

a	b	c	d	e	f	g	h	i	j	Classified as
90	0	2.5	0	5	0	2.5	0	0	0	a = Mouse
0	**97.5**	0	0	0	0	2.5	0	0	0	b = Scholar glue
0	0	**100**	0	0	0	0	0	0	0	c = Rubber duck
0	0	0	**90**	5	0	0	5	0	0	d = Smartphone
2.5	0	2.5	7.5	**85**	2.5	0	0	0	0	e = Bell pepper
0	0	0	0	0	**97.5**	0	2.5	0	0	f = Wool
0	0	5	0	0	0	**95**	0	0	0	g = Ball
0	0	0	2.5	0	2.5	0	**95**	0	0	h = Sponge
0	0	0	0	0	0	0	0	**100**	0	i = Can
2.5	5	0	0	0	0	0	0	0	**92.5**	j = Apple

4.3.5 Support Vector Machine (SVM)

The confusion matrix in Table 6 shows the SVM classifier results. It reveals that the model achieved a performance of 96.25% (385 hits) of instances correctly classified. In this case, the SVM is superior to *k*-NN by 2% and superior to DT by 8%. The classes best classified were Can and Apple, both achieving 100% performance. By contrast, the class Bell pepper achieved the lowest performance (90% of correct classifications).

Table 6 Confusion matrix for the Support Vector Machine algorithm. The classes Can and Apple achieved the best performance (100%).

a	b	c	d	e	f	g	h	i	j	Classified as
95	0	2.5	0	0	0	2.5	0	0	0	a = Mouse
0	**97.5**	0	0	0	0	2.5	0	0	0	b = Scholar glue
2.5	0	**95**	0	2.5	0	0	0	0	0	c = Rubber duck
0	0	0	**95**	2.5	0	0	2.5	0	0	d = Smartphone
2.5	0	0	7.5	**90**	0	0	0	0	0	e = Bell pepper
0	0	0	0	0	**97.5**	0	2.5	0	0	f = Wool
0	0	5	0	0	0	**95**	0	0	0	g = Ball
0	0	0	2.5	0	0	0	**97.5**	0	0	h = Sponge
0	0	0	0	0	0	0	0	**100**	0	i = Can
0	0	0	0	0	0	0	0	0	**100**	j = Apple

4.4 Performance Using ROC

Finally, we compared the models' performances for TPR, FPR and AUC to discard misclassification in negative classes. We recall that the point closest to the coordinate (0, 1) is best in the ROC curve, and the AUC should ideally be 1 for the best classification. Tables 7 – 16 show the results for the mentioned performance index.

Table 7 Performance values for the personal computer Mouse class.

Classifier	AUC	TPR	FPR
DT	0.908	0.800	0.031
NB	0.996	0.875	0.011
NN	**0.998**	**0.925**	**0.008**
K-NN	0.947	0.900	0.006
SVM	0.992	0.950	0.006

Table 8 Performance values for the Scholar glue class.

Classifier	AUC	TPR	FPR
DT	0.918	0.800	0.008
NB	**0.996**	**0.975**	**0.000**
NN	0.994	0.950	0.008
K-NN	0.985	0.975	0.006
SVM	0.994	0.975	0.000

Table 9 Performance values for the Rubber duck class.

Classifier	AUC	TPR	FPR
DT	0.918	0.800	0.014
NB	0.994	0.850	0.014
NN	**0.997**	**0.975**	**0.014**
K-NN	0.994	1.000	0.011
SVM	0.994	0.950	0.008

Table 10 Performance values for the Smartphone class.

Classifier	AUC	TPR	FPR
DT	0.953	0.925	0.008
NB	**0.995**	**0.925**	**0.011**
NN	0.995	0.900	0.014
K-NN	0.945	0.900	0.011
SVM	0.992	0.950	0.011

Table 11 Performance values for the Bell pepper class.

Classifier	AUC	TPR	FPR
DT	0.935	0.900	0.022
NB	**0.989**	**0.850**	**0.017**
NN	0.968	0.825	0.006
K-NN	0.920	0.850	0.011
SVM	0.985	0.900	0.006

Table 12 Performance values for the Wool class.

Classifier	AUC	TPR	FPR
DT	0.969	0.925	0.011
NB	**0.998**	**0.975**	**0.008**
NN	0.997	0.975	0.003
K-NN	0.985	0.975	0.006
SVM	0.997	0.975	0.000

Table 13 Performance values for the Ball class.

Classifier	AUC	TPR	FPR
DT	0.967	0.950	0.011
NB	0.992	0.950	0.006
NN	**0.997**	**0.925**	**0.006**
K-NN	0.972	0.950	0.006
SVM	0.985	0.950	0.006

Table 14 Performance values for the Sponge class.

Classifier	AUC	TPR	FPR
DT	0.960	0.925	0.006
NB	0.999	0.925	0.003
NN	**0.999**	**0.950**	**0.008**
K-NN	0.971	0.950	0.008
SVM	0.996	0.975	0.006

Table 15 Performance values for the soda Can class.

Classifier	AUC	TPR	FPR
DT	0.987	0.975	0.003
NB	**1.000**	**1.000**	**0.000**
NN	1.000	1.000	0.003
K-NN	**1.000**	**1.000**	**0.000**
SVM	**1.000**	**1.000**	**0.000**

Table 16 Performance values for the Apples class.

Classifier	AUC	TPR	FPR
DT	0.928	0.825	0.017
NB	1.000	1.000	0.006
NN	0.999	0.925	0.006
K-NN	0.963	0.925	0.000
SVM	**1.000**	**1.000**	**0.000**

4.5 General Analysis of Results

Section 4.3 and Section 4.4 provide a quantitative comparison of the five classification algorithms according to Accuracy, FPR, FPR and AUC. This comparison allowed us to measure our pressure descriptor's behaviour and its ability to separate different object classes. The classification performance ranged from 88.25% (DT) to 96.25% (SVM).

The results in the confusion matrices show that DT achieved the lowest classification performance, with a minimum of only 80% of instances correctly classified. The classes that achieved the worst classification performance were

Mouse, Scholar glue and the Rubber duck. On the contrary, the SVM classifier achieved the best classifications, all above 90% of correctly classified instances. In this case, the classes Apple and Can gave 100% performance.

The objects best classified by the algorithms were: the *Can*, with Decision Tree 97.5%, Naive Bayes 100%, Neural Network 100%, *k*-NN 100% and Support Vector Machine 100%, giving an average of 99.5% (average obtained among the five classifiers); the *Wool* with 96.5%; the *Sponge* and *Ball* with 94.5%; the *School glue* and *Apple* with 93.5%; the *Smartphone* with 92%; the *Mouse* with 89%; and finally, the class that gave the lowest average classification percentage was *Bell pepper* with 86.5%.

5 Conclusions

In this study, we have described a recognition pipeline for a tactile recognition system. This system used a robotic gripper for object manipulation armed with tactile sensors that allowed a Machine Learning algorithm to recognize objects with high accuracy and sensitivity.

Our recognition methodology required a training set of known objects obtained using the robotic gripper armed with a set of pressure sensors. The pipeline included the design of a pressure descriptor composed of ten voltage measurements retrieved from the pressure sensors using the methodology described in Section 3. The pressure descriptor allowed us to characterize rigid and flexible objects. The voltage variations from the tactile sensors were continuous values representing the local pressure applied to the object.

We used a small dataset with ten object classes during experiments, similar to dataset sizes used in previous work. This size allowed us to generate a minimal prototype and evaluate our method at an early stage. Even a few object classes can lead to a less generalizable model; we expect to reduce this gap by adding more objects in a future version of this database. We evaluated the pressure descriptor's benefits by training and testing a set of learning algorithms. The evaluation considered values of the TPR (True Positive Rate) and FPR (False Positive Rate) as performance measures, which correspond to the nearest point to the coordinate (0, 1) of the ROC curve. The results showed that the SVM achieved the best performance with a TPR = 95.5% and an FPR = 0.5%. These results are encouraging and indicate that recognizing objects using tactile sensing can be performed without human intervention. However, we acknowledge the implicit limitation in machine learning models for recognizing unknown objects. For example, the recognition algorithm will be unable to differentiate a pear if the dataset only contains apples or oranges as a training example. However, this kind of recognition system work is not for general operation. They typically have to analyze a limited set of objects when the system is in a real production environment. Currently, several production processes require manipulating objects to control their quality. For instance, companies that export fruit with a control line operated by humans can easily include this framework in their procedures, increasing the quality levels and objectivity in human inspection tasks.

We visualize three research paths for future work. Firstly, we can improve the recognition pipeline and the pressure descriptor, exploring the best means of characterizing new surfaces and improving identification performance.

Secondly, we propose to run tests using different sensors to evaluate the variability of the measurements using different hardware, *e.g.* a professional gripper for industrial use, that would allow us to validate the repeatability of the descriptor. In this case, we propose input level normalization to avoid scale changes in using different hardware configurations for data acquisition. This change allows the model to be independent of the acquisition stage. Thirdly, we propose extending the recognition pipeline, including a computer vision system. This last stage would allow us to use visual features to enhance recognition capabilities and help human decision-making in automatic product quality assurance.

Acknowledgments. This work was supported in part by DIUDA Grant No. 22406 and No. 22345 from Universidad de Atacama.

References

[1] Salmani-Ghabeshi, S., Palomo-Marín, M.R., Bernalte, E., Rueda-Holgado, F., Miró-Rodríguez, C., Cereceda-Balic, F., Fadic, X., Vidal, V., Funes, M., Pinilla-Gil, E.: Spatial gradient of human health risk from exposure to trace elements and radioactive pollutants in soils at the puchuncav´ı-ventanas industrial complex, Chile. Environmental Pollution 218, 322–330 (2016). https://doi.org/10.1016/j.envpol.2016.07.007

[2] Newman, T., Jain, A.K.: A survey of automated visual inspection. Computer Vision and Image Understanding 61, 231–262 (1995)

[3] Cui, H., Radosavljevic, V., Chou, F., Lin, T., Nguyen, T., Huang, T., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 2090–2096 (2019). https://doi.org/10.1109/ICRA.2019.8793868

[4] Paden, B., Cap, M., Yong, S.Z., Yershov, D., Frazzoli, E.: A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on Intelligent Vehicles 1(1), 33–55 (2016). https://doi.org/10.1109/TIV.2016.2578706

[5] Fernandes, L.C., Souza, J.R., Pessin, G., Shinzato, P.Y., Sales, D., Mendes, C., Prado, M., Klaser, R., Magalhães, A.C., Hata, A., Pigatto, D., Branco, K.C., Grassi, V., Osorio, F.S., Wolf, D.F.: CaRINA intelligent robotic car: architectural design and applications. Journal of Systems Architecture 60(4), 372–392 (2014). https://doi.org/10.1016/j.sysarc.2013.12.003

[6] Bai, L., Yang, J., Chen, X., Sun, Y., Li, X.: Medical robotics in bone fracture reduction surgery: a review. Sensors 19(16) (2019). https://doi.org/10.3390/s19163593

[7] Bonatti, J., Vetrovec, G., Riga, C., Wazni, O., Stadler, P.: Robotic technology in cardiovascular medicine. Nat Rev Cardiol 11(5),266–275 (2014). https://doi.org/10.1038/nrcardio.2014.23

[8] Martins, G.S., Santos, L., Dias, J.: User-adaptive interaction in social robots: a survey focusing on non-physical interaction. International Journal of Social Robotics 11(1), 185–205 (2019). https://doi.org/10.1007/ s12369-018-0485-4

[9] Sahoo, A., Dwivedy, S.K., Robi, P.S.: Advancements in the field of autonomous underwater vehicle. Ocean Engineering 181, 145–160 (2019). https://doi.org/10.1016/j.oceaneng.2019.04.011

[10] Sato, T., Kim, K., Inaba, S., Matsuda, T., Takashima, S., Oono, A., Takahashi, D., Oota, K., Takatsuki, N.: Exploring hydrothermal deposits with multiple autonomous underwater vehicles. In: 2019 IEEE Underwater Technology (UT), pp. 1–5 (2019). https://doi.org/10.1109/UT.2019. 8734460

[11] Takada, T., Fukunaga, T., Maekawa, T.: New method for gas identification using a single semiconductor sensor. Sensors and Actuators B: Chemical 66(1-3), 22–24 (2000). https://doi.org/10.1016/s0925-4005(99) 00404-9

[12] Wu, X., Li, A.G., Wu, D.F., Ma, Z.: Calibration of line structured light sensor for robotic inspection system. AMM 44-47, 702–706 (2010). https: //doi.org/10.4028/www.scientific.net/amm.44-47.702

[13] Drimus, A., Kootstra, G., Bilberg, A., Kragic, D.: Design of a flexible tactile sensor for classification of rigid and deformable objects. Robotics and Autonomous Systems 62(1), 3–15 (2014). https://doi.org/10.1016/j. robot.2012.07.021

[14] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)

[15] Mutlu, R., Alici, G., in het Panhuis, M., Spinks, G.M.: 3d printed flexure hinges for soft monolithic prosthetic fingers. Soft Robotics 3(3), 120–133 (2016). https://doi.org/10.1089/soro.2016.0026

[16] Belzile, B., Birglen, L.: A compliant self-adaptive gripper with proprioceptive haptic feedback. Autonomous Robots 36(1-2), 79–91 (2013). https://doi.org/10.1007/s10514-013-9360-1

[17] Romano, J.M., Hsiao, K., Niemeyer, G., Chitta, S., Kuchenbecker, K.J.: Human-inspired robotic grasp control with tactile sensing. IEEE Transactions on Robotics 27(6), 1067–1079 (2011). https://doi.org/10.1109/tro. 2011.2162271

[18] Denei, S., Maiolino, P., Baglini, E., Cannata, G.: On the development of a tactile sensor for fabric manipulation and classification for industrial applications. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Institute of Electrical and Electronics Engineers (IEEE), (2015). https://doi.org/10.1109/iros.2015.7354092

[19] Dwivedi, S.K., Vishwakarma, M., Soni, A.: Advances and researches on non destructive testing: a review. Materials Today: Proceedings 5(2), 3690–3698 (2018)

[20] Lawton J: The Role Of Robots In Industry 4.0. https://www.forbes.com/sites/jimlawton/2018/03/20/the-role-of-robots-in-industry-4-0/#7eb1d058706b. [Last accessed: April 30, 2019]

[21] Amazon Robotics: 2017 Amazon Robotics Challenge Official Rules. https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/arc/2017-amazon-robotics-challenge-rules-v3.pdf. [Last accessed: April 30, 2019] (2017)

[22] Amazon Robotics: Amazon Picking Challenge, Amazon Robot Research Project. https://amazonpickingchallenge.org/. [Last accessed: April 30, 2019]

[23] Girão, P.S., Ramos, P.M.P., Postolache, O., Pereira, J.M.D.: Tactile sensors for robotic applications. Measurement 46(3), 1257–1271 (2013)

[24] Park, J., Kim, M., Lee, Y., Lee, H.S., Ko, H.: Fingertip skin–inspired microstructured ferroelectric skins discriminate static/dynamic pressure and temperature stimuli. Science Advances 1(9), 1500661 (2015)

[25] De Rossi, D., Nannini, A., Domenici, C.: Artificial sensing skin mimicking mechanoelectrical conversion properties of human dermis. IEEE Transactions on Biomedical Engineering 35(2), 83–92 (1988)

[26] Ohtsuka, T., Furuse, A., Kohno, T., Nakajima, J., Yagyu, K., Omata, S.: Application of a new tactile sensor to thoracoscopic surgery: experimental and clinical study. Annals of Thoracic Surgery 60(3), 610–614 (1995)

[27] Sastry, S., Cohn, M., Tendick, F.: Milli-robotics for remote, minimally invasive surgery. Robotics and Autonomous Systems 21(3), 305–316 (1997)

[28] Kappassov, Z., Corrales, J.-A., Perdereau, V.: Tactile sensing in dexterous robot hands. Robotics and Autonomous Systems 74, 195–220 (2015)

[29] Lee, M.H.: Tactile sensing: new directions, new challenges. The International Journal of Robotics Research 19(7), 636–643 (2000)

[30] Zou, L., Ge, C., Wang, Z.J., Cretu, E., Li, X.: Novel tactile sensor technology and smart tactile sensing systems: a review. Sensors 17(11), 2653 (2017)

[31] Harmon, L.D.: Touch-sensing technology - a review. Society of Manufacturing Engineers, 1980. 58 (1980)

[32] Chi, C., Sun, X., Xue, N., Li, T., Liu, C.: Recent progress in technologies for tactile sensors. Sensors 18(4), 948 (2018)

[33] Hoelscher, J., Peters, J., Hermans, T.: Evaluation of tactile feature extraction for interactive object recognition. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 310–317 (2015). https://doi.org/10.1109/HUMANOIDS.2015.7363560

[34] Salzberg, S.L.: C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Machine Learning 16(3), 235–240 (1994). https://doi.org/10.1007/bf00993309

[35] John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345 (1995). Morgan Kaufmann Publishers Inc.

[36] Wilusz, T.: Neural networks – a comprehensive foundation. Neurocomputing 8(3), 359–360 (1995). https://doi.org/10.1016/0925-2312(95) 90026-8

[37] Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991). https://doi.org/10.1007/ bf00153759

[38] Esposito, F., Malerba, D., Semeraro, G., Kay, J.: A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5), 476–491 (1997). https://doi.org/ 10.1109/34.589207

[39] Theodoridis, S., Pikrakis, A., Koutroumbas, K., Cavouras, D.: Introduction to pattern recognition. Academic Press, Boston (2010)

[40] Bishop, C. M.: Pattern Recognition and Machine Learning. Springer (2006)

[41] Fawcett, T.: ROC graphs: notes and practical considerations for data mining researchers. (2003). Copyright Hewlett-Packard Company (2003)

[42] Zhang, Bing, et al. "Detection and identification of object based on a magnetostrictive tactile sensing system." IEEE Transactions on Magnetics 54.11 (2018): 1-5.

[43] Li, Guozhen, et al. "Skin-inspired quadruple tactile sensors integrated on a robot hand enable object recognition." Science Robotics 5.49 (2020): eabc8134.

[44] Mazid, Abdul Md, and R. Andrew Russell. "A robotic opto-tactile sensor for assessing object surface texture." 2006 IEEE Conference on Robotics, Automation and Mechatronics. IEEE, 2006.

[45] Yousef, Hanna, Mehdi Boukallel, and Kaspar Althoefer. "Tactile sensing for dexterous in -hand manipulation in robotics- a review." Sensors and Actuators A: Physical 167.2 (2011): 171-187.