Official Journal of the British Institute of Non-Destructive Testing

INSIGHT

Vol 64 | No 7
July 2022

Non-Destructive Testing and Condition Monitoring

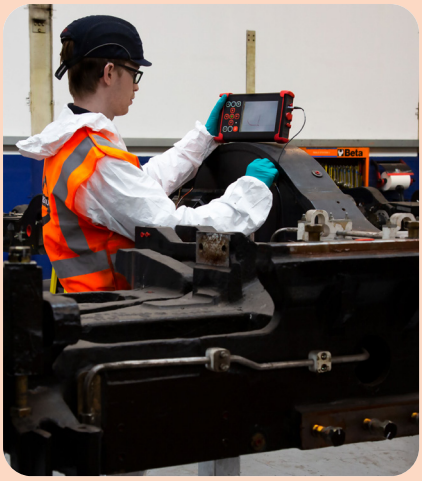
Official Journal of the British Institute of Non-Destructive Testing

INSIGHT

Non-Destructive Testing and Condition Monitoring

Volume 64 Number 7 July 2022

This month's cover



This month's front cover features the WeldCheck+ from ETher NDE being used at Transport for London (TFL), Acton, where it is employed to inspect seam welds on the bogies from the 2009 tube stock. As part of providing key transport services in the day-to-day running of the capital city, non-destructive testing (NDT) inspection is a key part of the standard engineering and safety procedures, where around a dozen personnel make up the NDT inspection team at the Acton base.

The WeldCheck eddy current testing (ECT) flaw detector family from ETher NDE appeals to users across a wide range of industries for coated weld inspection applications. The WeldCheck+ enhances an operator's inspection performance, with dual-frequency capability, automatic lift-off gain compensation (LOGC) and conductivity measurement.

ETher NDE's headquarters are located in Hertfordshire, UK, and due to the manufacturing and calibration teams based there, the company is able to build and ship its core eddy current flaw detectors and probes to meet customer requirements, either ex-stock or within two to three weeks of orders being placed.

ETher NDE
www.ethernde.com

Published by:
The British Institute of Non-Destructive Testing
Midsummer House, Riverside Way, Bedford Road
Northampton NN1 5NX, UK
Tel: +44 (0)1604 438300 | Fax: +44 (0)1604 438301
Email: info@bindt.org | Web: www.bindt.org

Features

Regular Features

DOI: 10.1784/insi.2022.64.7.383

Object recognition using tactile sensing in a robotic gripper

V Riffo, C Pieringer, S Flores and C Carrasco

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, for example visual recognition. In addition to a robot's ability to grasp objects without damaging them, it is also helpful to provide these machines with the ability to recognise objects while gently manipulating them, as humans do in the absence of or complementary to other senses. Advances in sensory technology have allowed for the accurate detection of different types of environment; however, the challenge of being able to efficiently represent sensory information persists. In this paper, a sensory system is proposed that allows a robotic gripper armed with pressure sensors to recognise objects through tactile manipulation. A pressure descriptor is designed to characterise the voltage magnitudes across different objects and, finally, machine learning algorithms are used to recognise each object category. The results show that the pressure descriptor characterises the different classes of objects in this experimental set-up. This system can complement other sensory data to perform different tasks in a robotic environment and future research areas are proposed to handle problems with tactile manipulation.

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 require palpations for quality assurance when detecting irregularities or damages, for example in the classification of fruit maturity states. Furthermore, other processes are dangerous and involve risk to life, for example the handling of radioactive materials[1]. In general, human performance in inspection tasks can decrease as a result of stress, fatigue or illness, among other factors[2]. This fact illustrates the point that robotic manipulation that allows for recognition and sensing of 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 automation of the processes. The authors believe that progress in this research area would be helpful to expand the usage 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, 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 the human abilities of sensing and decision-making, *ie* the robot's sensors can acquire stimuli from the environment and make decisions as humans do through their smell, vision and skin. The improvement and development of sensing technologies have helped robots become more precise in sensing various stimuli perceived by humans, for example in gas detectors[11], light sensors[12] and pressure sensors[13].

This work describes a methodology for recognising 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. In this paper, a series of voltage measurements are collected during data acquisition while the gripper puts pressure on the test object. Then, in the process pipeline, the system transforms the voltage levels into a feature descriptor to characterise the category of each object. This descriptor is designed to represent the voltage variations. This data is used as the input for a set of classification algorithms: decision tree (DT), naive Bayes (NB), neural network (NN), *k*-nearest neighbour (*k*NN) and support vector machine (SVM). These algorithms are all implemented using the Python module Scikit-learn[14]. This method allows for object recognition with different types of harness. Along with this research, two problems are addressed: the recognition of objects using pressure sensors, emulating human execution, and the automation of inspection processes, eventually providing an analytic tool for people working on production lines, with manual inspection providing quality indices, such as with fruit-exporting companies.

A literature review on tactile manipulation and recognition identifies three main research areas: the construction and performance of grippers[15,16], pressure sensors[17,18] and, finally, object recognition through grippers[13]. In the latter research, the use of the *k*NN algorithm provides the best results. In this study, the work in[13] is extended, incorporating another type of sensor and another set of machine learning algorithms. The results show that these additional changes improve the performance in recognition.

The experiments are carried out with a limited database (similar to the database used in[13]) and the authors manually collect and

● ***Submitted 14.11.21 / Accepted 24.05.22***

Vladimir Riffo, Sebastián Flores and Carlos Carrasco are with the Department of Informatics Engineering and Computer Sciences (DIICC), Universidad de Atacama, Avenida Copayapu 485, Copiapó, Atacama, Chile. Email: sebastian.flores@uda.cl / carlos.carrasco@alumnos.uda.cl*

Christian Pieringer is not currently affiliated with any institution. Email: christian.pieringer@gmail.com

**Corresponding author. Email: vladimir.riffo.b@gmail.com*

assign labels to ten objects: a computer mouse, a glue container, a rubber duck, a smartphone, a bell pepper, a ball of wool, a stress ball, a dishwashing sponge, a can of soda and an apple. All of these objects present a wide variability, from rigid to flexible. The true positive rate (TPR) and the false positive rate (FPR) performance indices are used to quantify and evaluate the best settings for the methodology. The results show the best performance in recognition of two classes using the SVM classifier: soda cans, TPR = 100% and FPR = 0%, and apples, TPR = 100% and FPR = 3%. These results are encouraging and support the use of this 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 organised as follows. Section 2 presents related work on tactile sensors and concepts involved in this work. Section 3 introduces and describes the approach theoretically and summarises the classification method used in the experimental test. Section 4 describes the experimental set-up and the results of using this method. Finally, Section 5 outlines the 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 its customers[19]. In certain situations, these tests involve the manipulation of objects by human inspectors. Currently, an increasing number of companies are relying on automation in the inspection process. Robotic manipulators are the preferred mechanisms to mimic the human role during the inspection and manipulation[20-23]; however, they 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, such as temperature, shape and texture, through physical touch. These sensors allow the loop to be closed between the manipulator and the decision system. In the last 40 years, research on tactile sensing has demonstrated its 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 sensor present a particular behaviour of hysteresis that introduces noise during measurement. 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, the inspection sequence and movements and the processing techniques. In 2006, Mazid and Russell[44] introduced an opto-tactile sensor able to assess the surface texture of objects as an input to object recognition stages. 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 discuss using the sensor output to ingest recognition stages of classification models. In contrast, the current 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 them. The authors used a small dataset of 12 classes of objects. The results showed successful sensor measurements in a haptic-based object-classification scenario compared with a well-known industrial sensor. Finally, a *k*NN model discriminated between the time-series of the different pressure patterns. In the current work, the authors use a similar set-up 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 recognise 49 real-world 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 types of stiffness, repeating the grasp 30 times for each material. The 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. The current work expands on the representation the object, including different palpitation points.

Recently in[43], Li *et al* proposed a quadruple skin-inspired 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 recognise objects precisely. The system also included a stage of data integration through machine learning techniques. In this work, the authors focus on a way of representing the pressure vector. Instead of using multiple sensors, multiple rotations of the robotic hand are included. A small dataset of objects based on previous work is also used in order to present a prototype of this framework.

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

3. Method

This section describes the proposed methodology, including the pressure descriptor based on the tactile sensor measurements, as follows: overview, pressure descriptors, characterisation and training, classification algorithms and the receiver operating characteristic.

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

3.1 Overview

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

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

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 class of the object according to a machine learning model previously trained using a training set. Using this pipeline, the authors manually collected and built a dataset with pressure measurements from ten objects. The gripper acquired 40 different pressure points during this process by rotating the object for each sample (see Figure 2).

3.2 Pressure descriptor

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

The stepper motor performed micro-steps to obtain the pressure descriptor, causing the fingers of the gripper 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, *ie* 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.

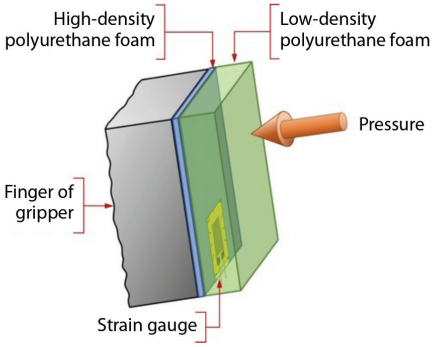


Figure 3. Details of one of the fingers of the gripper

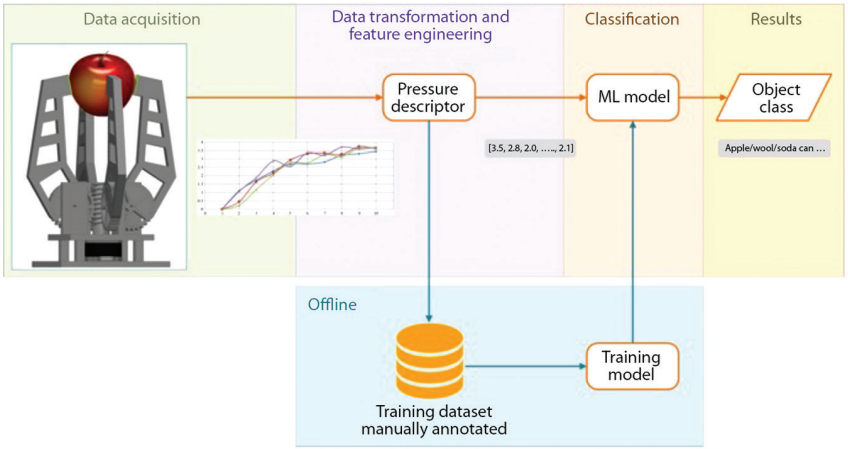


Figure 1. Diagram of the processing pipeline. It has three stages that range from data acquisition to classification. Training of the models was performed offline and used a manually collected dataset

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

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

where c corresponds to the class of the object, which varies 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 varies according to the pressure exerted on an object of class c and in the position i.

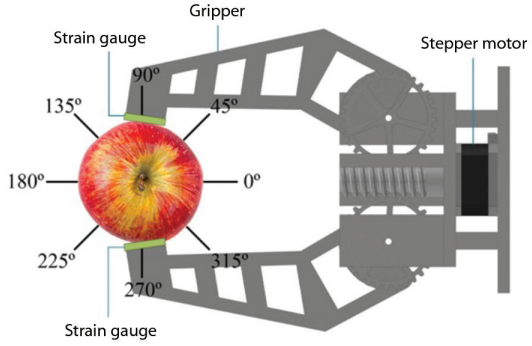


Figure 2. Diagram of the process of characterisation of objects

3.3 Characterisation and training

Each object was manually placed in 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, in the case of the apple, as seen in Figure 2, the gripper obtained 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, two of the four gripper fingers were removed, 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 $j = 1, \ldots, 10$, the respective pressure values

were collected, as shown in Table 1 and Figure 2. This procedure was repeated for each object, collecting the pressure levels and annotating the object label.

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

Angle (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

3.4 Classification algorithms

The dataset formation and model training were an 'offline' procedure. A set of machine learning algorithms used for classification were implemented: decision tree[34], naive Bayes[35], neural network[36], k-nearest neighbour and support vector machine[37]. These five algorithms were used to evaluate the discriminative power of the pressure descriptor and find the classifier that performed best in the processing pipeline for the different object flexibility and stiffness states. In the following subsections, the classifiers included in this study are briefly described.

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 work as a decision process. The tree begins at a node called the root, which 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 structure of the tree while keeping the interpretability without compromising accuracy[34], for example 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 $x = [x(1), \ldots, x(l)]^T \in \mathbb{R}^l$, given by:

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

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 probability density function. Despite the naive assumption of the feature independence given the class, the overall performance is reasonable since reliable estimates of the one-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 function of the human brain. A simple ANN model is the perceptron. Each perceptron is excited by a weighted sum over the input signals $x(1)$, $x(2)$, …, $x(l)$ with their corresponding weights w_1, w_2, …,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', *ie* 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 minimising a cost function. The most widely used option is the least-squares loss function:

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

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 minimisation. 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 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 kNN algorithm performs the following steps:

- Choose a value for k.
- Find the distance between x and all training points x_i, $i = 1, 2, \ldots, N$. Any measure of distance can be used (for example Euclidean or Mahalanobis).
- Find the k-nearest points to x.
- Compute the volume $V(x)$ in which the k-nearest neighbours lie.
- Compute the estimate by:

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

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

$$V(x) = 2\rho \text{ in one-dimensional space} \quad \ldots\ldots (5)$$

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

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

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 is another error-based model that aims to solve a non-linear classification task, mapping the feature vectors into a larger-dimensional space. The classes are expected to be linearly separable. This mapping is given by:

$$x \mapsto \phi(x) \in H \quad \text{..... (8)}$$

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 the mapping function is carefully chosen from a known family of functions with desirable and specific properties, the inner product between the images ($\phi(x_1)$ and $\phi(x_2)$) of two points x_1 and x_2 can be written as:

$$< \phi(x_1), \phi(x_2) > = k(x_1, x_2) \quad \text{..... (9)}$$

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 optimisation 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 set-up used to conduct the experiments to validate the methodology. This set-up is a prototype for this research and future work in this area could include improvements to the set-up.

4.1 Experimental set-up

An acquisition system was developed to evaluate the methodology (see Figure 4). This pipeline starts with a 3D-printed gripper designed by the authors, motorised by a 5 V 1.8° stepper motor, model KHP-11M10, manufactured by OKI Electric. This motor has high torque and tiny steps, which favours the palpation of objects, *ie* as smoothly as a human hand can. A 5 V 1000 mA power supply energises a Toshiba ULN2803APG that drives and protects the coils of the stepper motor. In a Wheatstone bridge configuration, the gripper has attached a set of four strain gauges of 3.5 mm and 120 ohms as a pressure sensor. Two of the four strain gauges R_1 and R_2 are attached to two of the 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. An INA128P instrumentation amplifier from Texas Instruments 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.

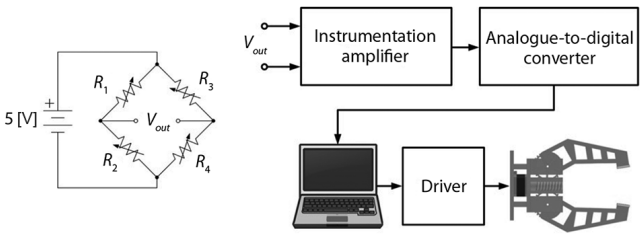


Figure 4. Data acquisition system

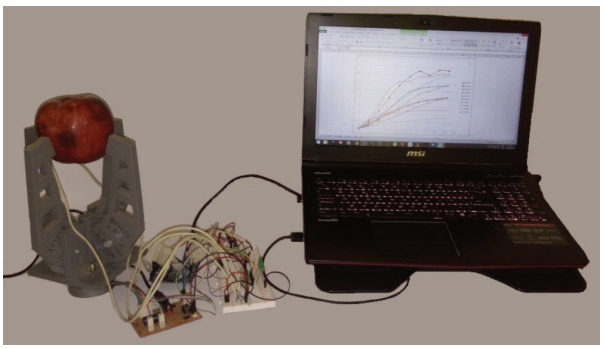


Figure 5. System used for object inspection

4.2 Dataset description

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

Figure 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 visualises the data in Table 1 to show how the pressure descriptor was created graphically. Not all of the objects used had a spherical or cylindrical shape. In such cases, for example the smartphone, the authors proceeded to palpate the four points randomly, taking care to ensure that none of these points coincided.

The pressure descriptor allowed the different objects used in this research to be characterised (see Figure 6). Using the acquired measurements and following the representation of Equation (1) gives:

$$F_{ct} = \left[f_j^{(i)}\right] = \begin{bmatrix} 0.020 & 1.104 & 1.733 & 2.290 & 2.705 & 2.705 & 2.827 & 3.223 & 3.315 & 3.452 \\ 0.005 & 0.449 & 1.636 & 2.163 & 2.944 & 3.311 & 3.350 & 3.276 & 3.740 & 3.638 \\ 0.005 & 0.234 & 1.182 & 2.056 & 2.788 & 2.769 & 3.262 & 3.149 & 3.647 & 3.687 \\ 0.005 & 1.074 & 1.890 & 2.881 & 2.568 & 3.369 & 3.223 & 3.716 & 3.608 & 3.691 \end{bmatrix} \quad \text{... (10)}$$

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

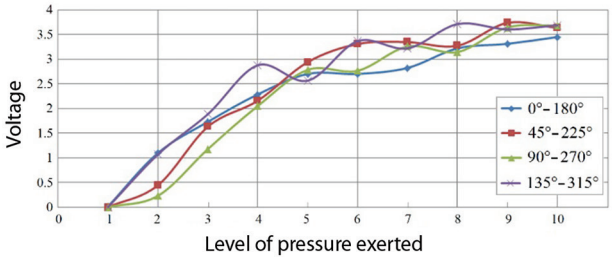


Figure 7. Graphical representation of the descriptor obtained from the pressure exerted on an apple

4.3 Evaluation and results

All of the experiments were conducted using Python for both of the main tasks of the pipeline: data acquisition and data classification. All of the classification algorithms were implemented using the Python module Scikit-learn[14].

Evaluation of the classification algorithm considered the standard performance indicators true positive rate and specificity or false positive rate, both widely used in the machine learning literature[40] and defined as follows:

$$\mathrm{TPR} = \frac{TP}{N_p} = \frac{TP}{TP + FN} \quad \ldots\ldots\ldots (11)$$

$$\mathrm{FPR} = \frac{FP}{N_n} = \frac{FP}{FP + FN} \quad \ldots\ldots\ldots (12)$$

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) summarises the behaviour of the TPR and FPR[41]. In order to take advantage of the complete dataset, experiments were carried out by applying the cross-validation technique using ten folds (90% of the dataset used for training and 10% used for testing). Further details about model evaluation techniques can be found in[40].

4.3.1 Decision tree

Confusion matrix analysis (see Table 2) shows that decision trees achieved an average accuracy of 88.25%. The can of soda achieved the best classification, obtaining 97.5% of correct classifications. By contrast, the computer mouse, glue container and rubber duck gave the lowest performance (80%).

4.3.2 Naive Bayes

For the naive Bayes classifier, the confusion matrix results indicate that the algorithm achieved an average accuracy of 93.25% (see Table 3). This performance is higher than decision trees by 5%. Specifically, the can of soda and the apple achieved the best performance (100%) while the rubber duck and the bell pepper gave the poorest performance, achieving 85% of correct classifications.

4.3.3 Neural network

The analysis carried out on the confusion matrix showed that the NN correctly classified an average of 93.5% of the instances, which is slightly higher than naive Bayes by 0.25% (see Table 4). The can of soda achieved the best performance (100%). The lowest performance for this model occurred for the bell pepper, which only achieved 82.5% of instances correctly classified.

4.3.4 k-nearest neighbours

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

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 = computer mouse
10	**80**	0	0	0	0	0	0	0	10	b = glue container
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 = ball of wool
2.5	0	2.5	0	0	0	**95**	0	0	0	g = stress ball
0	0	0	2.5	0	5	0	**92.5**	0	0	h = dishwashing sponge
0	0	0	0	0	2.5	0	0	**97.5**	0	i = can of soda
7.5	7.5	0	0	0	0	2.5	0	0	**82.5**	j = apple

Table 3. Confusion matrix for the naive Bayes algorithm. The apple 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 = computer mouse
0	**97.5**	0	0	0	0	2.5	0	0	0	b = glue container
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 = ball of wool
2.5	0	2.5	0	0	0	**95**	0	0	0	g = stress ball
0	0	0	2.5	0	5	0	**92.5**	0	0	h = dishwashing sponge
0	0	0	0	0	0	0	0	**100**	0	i = can of soda
0	0	0	0	0	0	0	0	0	**100**	j = apple

4.3.5 Support vector machine

The confusion matrix in Table 6 shows the SVM classifier results. The model achieved a performance of 96.25% (385 hits) of instances correctly classified, being superior to the *k*NN by 2% and superior to the DT by 8%. The objects best classified were the can of soda and the apple, both achieving 100% performance. By contrast, the bell pepper achieved the lowest performance (90% of correct classifications).

4.4 Performance using ROC

Finally, the performances of the models were compared for TPR, FPR and AUC to discard misclassification in negative classes. It is recalled 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 4. Confusion matrix for the neural network algorithm. The can of soda 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 = computer mouse
0	**95**	0	0	0	0	2.5	0	0	2.5	b = glue container
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 = ball of wool
0	2.5	5	0	0	0	**92.5**	0	0	0	g = stress ball
0	0	0	2.5	0	2.5	0	**95**	0	0	h = dishwashing sponge
0	0	0	0	0	0	0	0	**100**	0	i = can of soda
2.5	5	0	0	0	0	0	0	0	**92.5**	j = apple

Table 5. Confusion matrix for the *k*NN algorithm. The model achieved the best performance for the rubber duck and the can of soda

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 = computer mouse
0	**97.5**	0	0	0	0	2.5	0	0	0	b = glue container
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 = ball of wool
0	0	5	0	0	0	**95**	0	0	0	g = stress ball
0	0	0	2.5	0	2.5	0	**95**	0	0	h = dishwashing sponge
0	0	0	0	0	0	0	0	**100**	0	i = can of soda
2.5	5	0	0	0	0	0	0	0	**92.5**	j = apple

Table 6. Confusion matrix for the support vector machine algorithm. The can of soda and the 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 = computer mouse
0	**97.5**	0	0	0	0	2.5	0	0	0	b = glue container
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 = ball of wool
0	0	5	0	0	0	**95**	0	0	0	g = stress ball
0	0	0	2.5	0	0	0	**97.5**	0	0	h = dishwashing sponge
0	0	0	0	0	0	0	0	**100**	0	i = can of soda
0	0	0	0	0	0	0	0	0	**100**	j = apple

4.5 General analysis of results

Sections 4.3 and 4.4 provide a quantitative comparison of the five classification algorithms according to accuracy, FPR, FPR and AUC. This comparison allowed the behaviour of the pressure descriptor and its ability to separate different object classes to be measured. The classification performance ranged from 88.25% (DT) to 96.25% (SVM).

The results in the confusion matrices show that the DT achieved the lowest classification performance, with a minimum of only 80% of instances correctly classified. The objects that achieved the worst classification performance were the computer mouse, the glue container 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 apple and the can of soda gave 100% performance.

The objects best classified by the algorithms were: the can of soda, 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 ball of wool with 96.5%; the dishwashing sponge and the stress ball with 94.5%; the glue container and the apple with 93.5%; the smartphone with 92%; the computer mouse with 89%; and, finally, the class that gave the lowest average classification percentage was the bell pepper with 86.5%.

5. Conclusions

In this study, a recognition pipeline for a tactile recognition system is described. This system used a robotic gripper for object manipulation armed with tactile sensors that allowed a machine learning algorithm to recognise objects with high accuracy and sensitivity.

The 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 rigid and flexible objects

Table 7. Performance values for the computer mouse

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 glue container

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

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

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

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 ball of wool

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 stress ball

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 dishwashing sponge

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 can of soda

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 apple

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**

to be characterised. The voltage variations from the tactile sensors were continuous values representing the local pressure applied to the object.

A small dataset with ten object classes was used during the experiments, similar to dataset sizes used in previous work. This size allowed the generation of a minimal prototype and evaluation of the method at an early stage. Even a few object classes can lead to a less generalisable model; it is expected that this gap will be reduced by adding more objects in a future version of this database. Benefits of the pressure descriptor were evaluated by training and testing a set of learning algorithms. The evaluation considered values of the TPR and FPR 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 of 95.5% and an FPR of 0.5%. These results are encouraging and indicate that recognising objects using tactile sensing can be performed without human intervention. However, the implicit limitation in machine learning models for recognising unknown objects is acknowledged.

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 type of recognition system work is not for general operation. Typically, a limited set of objects have to be analysed when the system is in a real production environment. Currently, several production processes require the manipulation of objects to control their quality. For instance, companies that export fruit with a production line operated by humans can easily include this framework in their procedures, increasing the quality levels and objectivity in human inspection tasks.

Three research paths are visualised for future work. Firstly, the recognition pipeline and the pressure descriptor can be improved, exploring the best means of characterising new surfaces and improving identification performance. Secondly, the authors propose to run tests using different sensors to evaluate the variability of the measurements using different hardware, for example a professional gripper for industrial use, which would allow the repeatability of the descriptor to be validated. In this case, input level normalisation is proposed to avoid scale changes when using different hardware configurations for data acquisition. This change allows the model to be independent of the acquisition stage. Thirdly, extention of the recognition pipeline is proposed, including a computer vision system. This last stage would allow visual features to be used to enhance recognition capabilities and aid human decision-making in automatic product quality assurance.

Acknowledgements

This work was supported in part by DIUDA, Universidad de Atacama, Chile (grant numbers 22406 and 22345).

References

1. S Salmani-Ghabeshi, M R Palomo-Marín, E Bernalte, F Rueda- Holgado, C Miró-Rodríguez, F Cereceda-Balic, X Fadic, V Vidal, M Funes and E Pinilla-Gil, 'Spatial gradient of human health risk from exposure to trace elements and radioactive pollutants in soils at the Puchuncaví-Ventanas industrial complex, Chile', Environmental Pollution, Vol 218, pp 322-330, 2016. DOI: 10.1016/j.envpol.2016.07.007
2. T Newman and A K Jain, 'A survey of automated visual inspection', Computer Vision and Image Understanding, Vol 61, No 2, pp 231-262, 1995.
3. H Cui, V Radosavljevic, F Chou, T Lin, T Nguyen, T Huang, J Schneider and N Djuric, 'Multi-modal trajectory predictions for autonomous driving using deep convolutional networks', Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, pp 2090-2096, 20-24 May 2019. DOI: 10.1109/ICRA.2019.8793868
4. B Paden, M Cap, S Z Yong, D Yershov and E Frazzoli, 'A survey of motion planning and control techniques for self-driving urban vehicles', IEEE Transactions on Intelligent Vehicles, Vol 1, No 1, pp 33-55, 2016. DOI: 10.1109/TIV.2016.2578706
5. L C Fernandes, J R Souza, G Pessin, P Y Shinzato, D Sales, C Mendes, M Prado, R Klaser, A C Magalhães, A Hata, D Pigatto, K C Branco, V Grassi, F S Osorio and D F Wolf, 'CaRINA intelligent robotic car: architectural design and applications', Journal of Systems Architecture, Vol 60, No 4, pp 372-392, 2014. DOI: 10.1016/j.sysarc.2013.12.003
6. L Bai, J Yang, X Chen, Y Sun and X Li, 'Medical robotics in bone fracture reduction surgery: a review', Sensors, Vol 19, No 16, 2019. DOI: 10.3390/s19163593
7. J Bonatti, G Vetrovec, C Riga, O Wazni and P Stadler, 'Robotic technology in cardiovascular medicine', Nature Reviews Cardiology, Vol 11, No 5, pp 266-275, 2014. DOI: 10.1038/nrcardio.2014.23
8. G S Martins, L Santos and J Dias, 'User-adaptive interaction in social robots: a survey focusing on non-physical interaction', International Journal of Social Robotics, Vol 11, No 1, pp 185-205, 2019. DOI: 10.1007/s12369-018-0485-4
9. A Sahoo, S K Dwivedy and P S Robi, 'Advancements in the field of autonomous underwater vehicle', Ocean Engineering, Vol 181, pp 145-160, 2019. DOI: 10.1016/j.oceaneng.2019.04.011
10. T Sato, K Kim, S Inaba, T Matsuda, S Takashima, A Oono, D Takahashi, K Oota and N Takatsuki, 'Exploring hydrothermal deposits with multiple autonomous underwater vehicles', Proceedings of the 2019 IEEE Underwater Technology (UT) Conference, Kaohsiung, Taiwan, pp 1-5, 16-19 April 2019. DOI: 10.1109/UT.2019.8734460
11. T Takada, T Fukunaga and T Maekawa, 'New method for gas identification using a single semiconductor sensor', Sensors and Actuators B: Chemical, Vol 66, No 1-3, pp 22-24, 2000. DOI: 10.1016/s0925-4005(99)00404-9
12. X Wu, A G Li, D F Wu and Z Ma, 'Calibration of line-structured light sensor for robotic inspection system', Applied Mechanics and Materials, Vol 44-47, pp 702-706, 2010. DOI: 10.4028/www.scientific.net/amm.44-47.702
13. A Drimus, G Kootstra, A Bilberg and D Kragic, 'Design of a flexible tactile sensor for classification of rigid and deformable objects', Robotics and Autonomous Systems, Vol 62, No 1, pp 3-15, 2014. DOI: 10.1016/j.robot.2012.07.021
14. F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot and E Duchesnay, 'Scikit-learn: machine learning in Python', Journal of Machine Learning Research, Vol 12, pp 2825-2830, 2011.
15. R Mutlu, G Alici, M in het Panhuis and G M Spinks, '3D printed flexure hinges for soft monolithic prosthetic fingers', Soft Robotics, Vol 3, No 3, pp 120-133, 2016. DOI: 10.1089/soro.2016.0026
16. B Belzile and L Birglen, 'A compliant self-adaptive gripper with proprioceptive haptic feedback', Autonomous Robots, Vol 36, No 1-2, pp 79-91, 2013. DOI: 10.1007/s10514-013-9360-1
17. J M Romano, K Hsiao, G Niemeyer, S Chitta and K J Kuchenbecker, 'Human-inspired robotic grasp control with tactile sensing', IEEE Transactions on Robotics, Vol 27, No 6, pp 1067-1079, 2011. DOI: 10.1109/tro.2011.2162271
18. S Denei, P Maiolino, E Baglini and G Cannata, 'On the development of a tactile sensor for fabric manipulation and classification for industrial applications', Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September-2 October 2015. DOI: 10.1109/iros.2015.7354092
19. S K Dwivedi, M Vishwakarma and A Soni, 'Advances and researches on non-destructive testing: a review', Materials Today: Proceedings, Vol 5, No 2, pp 3690-3698, 2018.
20. J Lawton, 'The role of robots in Industry 4.0'. Available at: www.forbes.com/sites/jimlawton/2018/03/20/the-role-of-robots-in-industry-4-0/#7eb1d058706b (Accessed: 30 April 2019).
21. Amazon Robotics, '2017 Amazon Robotics Challenge Official Rules', www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/arc/2017-amazon-robotics-challenge-rules-v3.pdf (Last accessed: 30 April 2019), 2017.

22. Amazon Robotics, 'Amazon Picking Challenge', Amazon Robot Research Project, https://amazonpickingchallenge.org/ (Last accessed: 30 April 2019).
23. P S Girão, P M P Ramos, O Postolache and J M D Pereira, 'Tactile sensors for robotic applications', Measurement, Vol 46, No 3, pp 1257-1271, 2013.
24. J Park, M Kim, Y Lee, H S Lee and H Ko, 'Fingertip skin-inspired microstructured ferroelectric skins discriminate static/dynamic pressure and temperature stimuli', Science Advances, Vol 1, No 9, 1500661, 2015.
25. D De Rossi, A Nannini and C Domenici, 'Artificial sensing skin mimicking mechanoelectrical conversion properties of human dermis', IEEE Transactions on Biomedical Engineering, Vol 35, No 2, pp 83-92, 1988.
26. T Ohtsuka, A Furuse, T Kohno, J Nakajima, K Yagyu and S Omata, 'Application of a new tactile sensor to thoracoscopic surgery: experimental and clinical study', Annals of Thoracic Surgery, Vol 60, No 3, pp 610-614, 1995.
27. S Sastry, M Cohn and F Tendick, 'Milli-robotics for remote, minimally invasive surgery', Robotics and Autonomous Systems, Vol 21, No 3, pp 305-316, 1997.
28. Z Kappassov, J-A Corrales and V Perdereau, 'Tactile sensing in dexterous robot hands', Robotics and Autonomous Systems, Vol 74, pp 95-220, 2015.
29. M H Lee, 'Tactile sensing: new directions, new challenges', The International Journal of Robotics Research, Vol 19, No 7, pp 636-643, 2000.
30. L Zou, C Ge, Z J Wang, E Cretu and X Li, 'Novel tactile sensor technology and smart tactile sensing systems: a review', Sensors, Vol 17, No 11, 2653, 2017.
31. L D Harmon, 'Touch-sensing technology – a review', SME Technical Report, 58 pp, Society of Manufacturing Engineers, 1980.
32. C Chi, X Sun, N Xue, T Li and C Liu, 'Recent progress in technologies for tactile sensors', Sensors, Vol 18, No 4, 948, 2018.
33. J Hoelscher, J Peters and T Hermans, 'Evaluation of tactile feature extraction for interactive object recognition', Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, South Korea, pp 310-317, 3-5 November 2015. DOI: 10.1109/HUMANOIDS.2015.7363560
34. S L Salzberg, 'C4.5: Programs for Machine Learning by J Ross Quinlan. Morgan Kaufmann Publishers, Inc, 1993', Book Review, Machine Learning, Vol 16, No 3, pp 235-240, 1994. DOI: 10.1007/bf00993309
35. G H John and P Langley, 'Estimating continuous distributions in Bayesian classifiers', Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp 338-345, Morgan Kaufmann Publishers Inc, 1995.
36. T Wilusz, 'Neural networks – a comprehensive foundation', Neurocomputing, Vol 8, No 3, pp 359-360, 1995. DOI: 10.1016/0925-2312(95)90026-8
37. D W Aha, D Kibler and M K Albert, 'Instance-based learning algorithms', Machine Learning, Vol 6, No 1, pp 37-66, 1991. DOI: 10.1007/bf00153759
38. F Esposito, D Malerba, G Semeraro and J Kay, 'A comparative analysis of methods for pruning decision trees', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 19, No 5, pp 476-491, 1997. DOI: 10.1109/34.589207
39. S Theodoridis, A Pikrakis, K Koutroumbas and D Cavouras, Introduction to Pattern Recognition, Academic Press, Boston, Massachusetts, USA, 2010.
40. C M Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
41. T Fawcett, 'ROC graphs: notes and practical considerations for data-mining researchers', Copyright Hewlett-Packard Company (2003), 2003.
42. B Zhang *et al*, 'Detection and identification of object based on a magnetostrictive tactile sensing system', IEEE Transactions on Magnetics, Vol 54, No 11, pp 1-5, 2018.
43. G Li *et al*, 'Skin-inspired quadruple tactile sensors integrated on a robot hand enable object recognition', Science Robotics, Vol 5, No 49, eabc8134, 2020.
44. A Md Mazid and R A Russell, 'A robotic opto-tactile sensor for assessing object surface texture', Proceedings of the 2006 IEEE Conference on Robotics, Automation and Mechatronics, Luoyang, China, 25-28 June 2006.
45. H Yousef, M Boukallel and K Althoefer, 'Tactile sensing for dexterous in-hand manipulation in robotics – a review', Sensors and Actuators A: Physical, Vol 167, No 2, pp 171-187, 2011.