

Research Article

Synthetic Sensor Array Training Sets for Neural Networks

Oded Medina , Roi Yozevitch, and Nir Shvalb 

Engineering Faculty, Ariel University, Ariel, Israel

Correspondence should be addressed to Nir Shvalb; nirsh@ariel.ac.il

Received 11 June 2019; Accepted 10 August 2019; Published 10 September 2019

Academic Editor: Franz L. Dickert

Copyright © 2019 Oded Medina et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is often hard to relate the sensor's electrical output to the physical scenario when a multidimensional measurement is of interest. An artificial neural network may be a solution. Nevertheless, if the training data set is extracted from a real experimental setup, it can become unreachable in terms of time resources. The same issue arises when the physical measurement is expected to extend across a wide range of values. This paper presents a novel method for overcoming the long training time in a physical experiment set up by bootstrapping a relatively small data set for generating a synthetic data set which can be used for training an artificial neural network. Such a method can be applied to various measurement systems that yield sensor output which combines simultaneous occurrences or wide-range values of physical phenomena of interest. We discuss to which systems our method may be applied. We exemplify our results on three study cases: a seismic sensor array, a linear array of strain gauges, and an optical sensor array. We present the experimental process, its results, and the resulting accuracies.

1. Introduction

Reconstruction problems are of interest to the research community due to their importance in engineering applications. These have been studied with regard to heat source identification [1–5], shape reconstruction [6], wave source identification [7–9], inverse boundary conditions reconstruction [10, 11], and the problem of force identification on plates [12–17]. A notable example is [14] by Chierichetti and Ruzzene which uses the *confluence algorithm* to numerically reconstruct the dynamic response field of the structure from a limited number of experimental measurements (for a survey on analytic methods, see [18]). Another reconstruction problem is considered in [19] where an optical sensor array is used to position an IR (reflected) light source.

This paper offers a new line of thought on such problems. We gather sensor inputs taken in simple scenarios and manipulate them in accordance with the physical governing equations to form synthetic data in order to reconstruct more involved scenarios.

A sensor measures a physical quantity or its derivatives (for example, the change of an electrical analog signal is converted into the corresponding physical unit which is of interest [20, 21]). We distinguish between two conceptual cases:

- (i) A sensor may be applied when a *single quantity* is to be detected. Examples include scalar quantities like velocity, temperature, distance, and light emission as the inputs to be estimated. Other examples are vector quantities like spatial orientation, detecting the location of a light source etc.
- (ii) A sensor may also be applied when *simultaneous quantities* are to be detected. Examples include scalar quantities like detecting all joint angles of a robotic arm [22], extracting the tensile stress field [6, 23], and measuring the motion of the sea bed [24]. Examples for simultaneous vector quantities to be detected include stress tensor field extraction and detecting the position of multiple light sources [25]

Using multiple sensors will not always detect simultaneous quantities. The reason for that is that in such cases, these quantities are aggregated into the sensor system—an *array of sensors*, in a complex manner. The same reasoning may apply also for the case where one is required to sense a wide range of a given physical quantity.

Artificial neural networks (ANN) are commonly used in computer vision [26] and regression problems. Though regression is a statistical process for estimating the relationships

among variables, it can be used for computing arbitrary functions. Being available and due to the increasing computing power, the ANN usage has been accelerated in almost every domain of computer science [27, 28]. A neural network may be a good solution to analyze the sensor data described in (i) to estimate a physical quantity [29]. Of course, one needs to obtain a sufficiently large training set for convergence (c.f. in [30]). Obviously for n simultaneous quantities to be detected, training the ANN would require to increase the training set exponentially in the power of n which is unrealizable in many cases.

The training neural network for image recognition, 3D data, and video is typically an expensive task of collecting and annotating big data sets. The task of automatically generating such labeled data is an emerging field of research; see for example [31–33]. A new different approach is called *domain randomization* [34–37] where synthetic data is created by adding random textures to real images in order to force the ANN to learn the essential features. The underlying logic of this paper is somewhat similar to this approach.

In this paper, we introduce a way to analyze data from a sensor array by *bootstrapping synthetic training data* for an ANN. This way, the ANN “decomposes” the simultaneous quantity inputs and estimates its components individually. We also show how one can bootstrap synthetic training data for an ANN to analyze a wide range of input values.

Remark. The term *bootstrap* has various interpretations in literature. In computer science, the term may indicate a procedure for simulating discrete events [38] or be used to indicate a way to iteratively improve a classifier’s performance [39]. In statistics, bootstrap refers to a computational manipulation in order to establish certain statistical values [40]. Some other fields of research use the term in a totally different meaning. Here, we use the term bootstrap to indicate a procedure in which one manipulates real data from sensor arrays to form artificial sensor array data.

1.1. Contribution and Paper Organization. The main concern of this paper is to construct synthetic physical measurements as data sets for ANNs. This is essential for cases where harvesting experimental data is out of reach (for reasons of time consumption, experimental setup complexity, etc.). For this end, we shall use basic ANNs provided by MATLAB.

The synthetic training data sets are bootstrapped out of a certain experimental set. More precisely, data is first collected from experiments; we then manipulate the sets to create a new training data. We discuss when such a method can be applied and exemplify our method for three cases of sensor arrays (cantilever deflection sensor, seismic sensors on a metal plate, and an optical sensor array).

This paper is organized as follows: Section 2 provides a mathematical formulation for situations where our method may be applied. Section 3 presents three sensor array study cases. Results for these cases where a single stimulus is of interest are provided in Section 5. In Section 6, we apply our bootstrapping method, and we conclude the paper in Section 7.

2. A Mathematical Formalism

In what follows, we shall examine situations where one may desire to generate synthetic training data for a neural network. For clarity, we shall consider three examples for application and follow them throughout the paper:

2.1. Study Case I. A horizontal cantilever beam with an unknown, nonconstant cross-section is included. A set of sensors (e.g., strain gauges) located along its axis, each on a different segment, measures the beam’s curvature (Figure 1). The beam is thought to be loaded by a *set of vertical forces* in which their position and magnitude are of interest. This sensor array was simulated using a finite element method. We used the finite element method (FEM) as a ground truth to train the ANN. Specifically, we used it to calculate the stress field and the resulting error function for training the ANN.

2.2. Study Case II. A set of seismic sensors (as depicted in Figure 2) is positioned peripherally on an the aluminum plate with an unknown, nonconstant thickness. The plate is subjected to a *single impulse* with unknown magnitude in which its position is of interest.

2.3. Study Case III. A light sensor array built out of 6 *light-dependent resistors* (LDR) which are situated on an arc as depicted in Figure 3 is included. A set of six lights in which their identity and magnitude are of interest, individually illuminates the sensor. We shall be interested in detecting up to two simultaneous light sources.

The commutative diagram in Figure 4 formulates the various scenarios discussed in this paper. We consider a *stimulus* x which is a map $x : C \rightarrow Y$ such that every sensor location $c \in C$ is mapped to a physical quantity. So, Y is the space of that measured physical quantity (typically, $Y \subset R$). For study case I, C is the set of stain sensor positions and Y is the curvature value range (or deflection). In study case II, C is the set of seismic sensor positions and Y is the plane’s peripheral movement space. In study case III, C is the set of points where the LDRs are situated and Y is the full range luminescence, measured, for example, in Lum.

The sensor’s transfer function $f : x \rightarrow M$ assigns an m dimensional electrical output in M to an m dimensional stimulus x (i.e., assigns a value to each sensor). The power map $f^n : x_1 \times x_2 \times \dots \times x_n \rightarrow M_1 \times M_2 \times \dots \times M_n$ assigns f for n experiments. For study case I, f computes the measured resistance in the i th segment as $R_i = \alpha x_i$ where α is a constant (explicitly, $R = \rho x_i / A$ where ρ is the resistivity of the sensor and A its cross-section) and x_i is the strain in the i th segment. For study case II, x_i is the amplitude and frequency of the plate in all sensor points due to an impulse of a unit magnitude (say, 37gr dropped from 20mm) and f computes the corresponding measured capacitance in all the seismic sensors. Note that since in case study II, we refer only to a single impulse, we are actually interested only in a diagonal subset $\text{diag}(x_1 \times x_2 \times \dots \times x_n)$ of $x_1 \times x_2 \times \dots \times x_n$ (which can be thought of as several impulses assigned to the same position). For study case

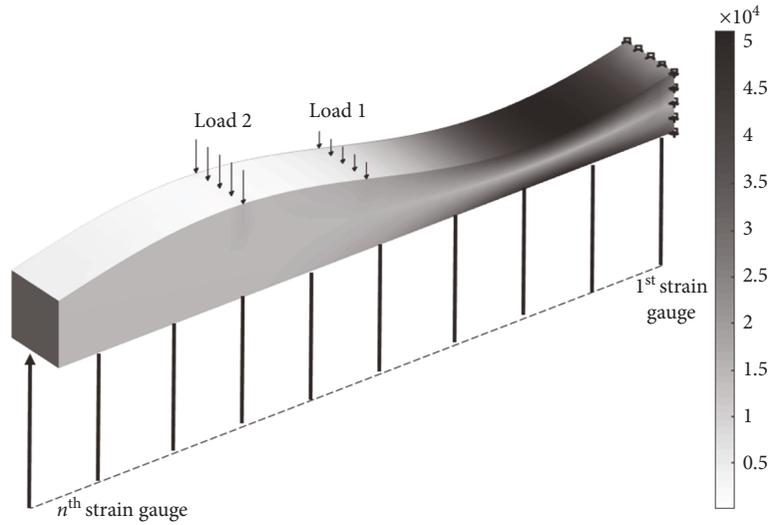


FIGURE 1: Study case I: experiment setup for the case where two loads are applied. The bottom arrows indicate the strain gauge-suggested positions. The gray scale map indicates the stress field given in Pascal. The 1-meter-length cantilever is assumed to be fixed at its right-hand side.

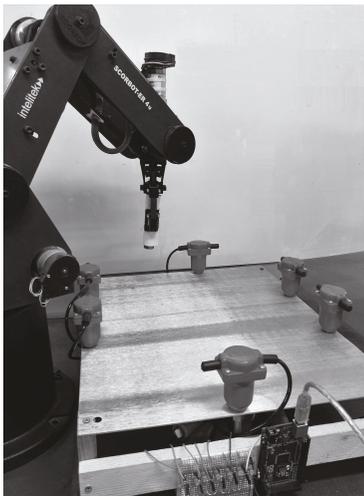


FIGURE 2: The six geophone sensors surrounding the plate. The robotic device repeatedly drops a brass rod on the plate.

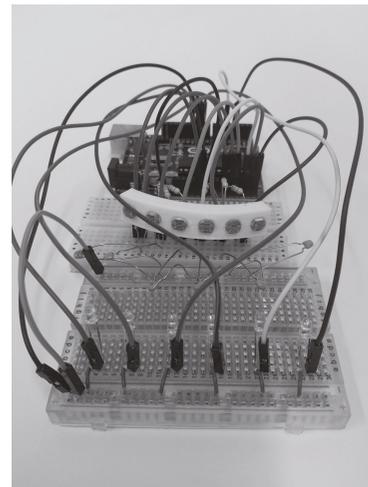


FIGURE 3: Study case III: experiment setup for the optical sensor array. Six LDRs are situated on a 50 mm radius arc of angle 60°. Six LEDs are positioned in front of the sensor array (not accurately aligned). At the first stage, LEDs were turned on individually with varying intensities. In the second stage, two light sources were turned on simultaneously with varying intensities.

III, the light sensor is based on LDRs. f is given by the LDR's datasheet which provides the relation

$$R = \frac{500}{L}, \tag{1}$$

between resistance R and luminous flux per unit area L , measured in Lux .

For all linear systems, the net response x of two or more simultaneous stimuli $\{x_i\}_{i=1}^n$ is simply their sum (i.e., g is the addition operator). Note that for the analysis which will be introduced here, the commutativity of the diagram in Figure 4 is enough. In the most general case, $g : x_1 \times x_2 \times \dots \times x_n \rightarrow x$ is the map (not necessarily linear) from the stimuli

$\{x_i\}_{i=1}^n$ to the net stimulus x (which is of the same dimension as a single x_i), for example, for study case I that would be stating the superposition principle for small deflections (in other words, saying that the strain in the i th segment can be summed up). For case II, a naive assumption would be to state superposition again but this is, of course, not the case as wave interference plays here an important role given that the metal plate is not uniform.

For study case III when two light sources project light on an LDR, its resistance changes due the resulting luminous flux per unit area equation (1). Evidently, for the case of k

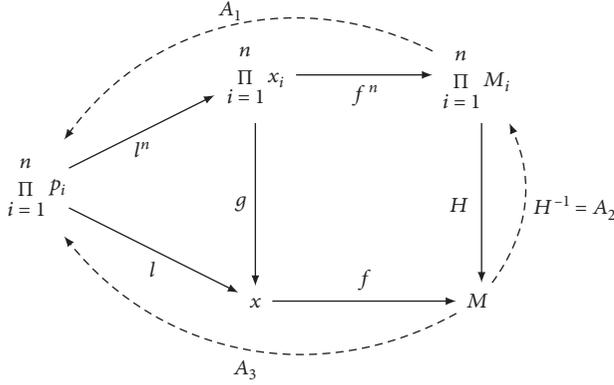


FIGURE 4: A commutative diagram representing the sensor arrays at hand.

light sources, the resistance is $R_{\text{total}} = 500/L_1 + L_2 + \dots + L_k$ with L_i as their respective detected luminance:

$$\frac{1}{R_{\text{total}}} = \frac{L_1}{500} + \frac{L_2}{500} + \dots = \sum_i \frac{1}{R_i}. \quad (2)$$

In other words, given a prior knowledge of the resistance due to each light source independently, one may calculate the resistance in a case where two light sources project their light simultaneously.

The phenomena which are of interest are marked $\{p_i\}_{i=1}^n$. For study case I, this corresponds to the amplitudes and positions of loads. The physical model ℓ is a mapping from the p_i 's to the stimulus x and for the case of simultaneous phenomena $\ell^n : p_1 \times p_2 \times \dots \times p_n \rightarrow x$. Note that there may be scenarios where ℓ may depend on the location of the sensors (e.g., nonisotropic plates, involving diodes etc.) In study case II, $\{p_i\}_{i=1}^n$ corresponds to dropping a unit magnitude weight. ℓ may be the transverse (shear) wave propagation principle while in study case I, this may refer to the Euler-Bernoulli beam deflection theory. For study case III, $\{p_i\}_{i=1}^n$ corresponds to the light intensity from the light sources. So, the map ℓ states the light propagation rules.

Finally, H is a map that assigns a set of separate set of outputs $\{M_i\}_{i=1}^n$ to a single output M . That is, H constructs a single synthetic sensor output from a set of known sensor outputs (that correspond to a set of stimuli resulting from the physical phenomena), under the assumption that the stimuli are simultaneous.

We assume the function H is a known decomposable multivariate function (e.g., H is a simple sum or some other decomposable functions as $H(f(x_1), f(x_2)) = f(x_1)f(x_2)/f(x_1) + f(x_2)$ as in [41], equation (1), which is exactly as in case III). A special case where a multivariate function H is a linear combination of products of functions in single variables was examined by Neuman in [42].

In many cases where the relevant physics is too complicated to compute or out of reach for any reason, one may use the *machine learning* approach to estimate the phenomena of interest. In numerous engineering scenar-

ios, the essential traits of the underlying measured object (e.g., deflected beam, vibrating plate, and sensor exact traits) cannot be attained, for example, (1) the inner geometry of the object contains nonnegligible holes and imperfections which cannot be inspected, (2) a portion of the object geometry is out of reach (e.g., pipe's inner wall, pressure vessels, and tightly connected plates), (3) objects which suffer slow corrosion and therefore their geometry changes over time, and (4) objects which their governing equations are not fully known but could be measured (e.g., wooden and composite material beams).

Explicitly, one may use an artificial neural network A_1 if a single-stimulus data set of real experiments is at hand. For cases where real experiments of some simultaneous stimuli are available, an ANN A_3 will do. There are cases where one can extract synthetic data using $H : \prod_{i=1}^n M_i \rightarrow M$ to use as input of A_3 without conducting a full set of experiments. In other cases, one may use M (i.e., real or synthetic data set) as an input to an ANN A_2 . This in turn is used to extract $\prod_{i=1}^n M_i$ which then can be used as input for A_1 .

3. Sensor Array Examples

We shall now introduce the first step of our training scheme, that is, we consider a situation where a single quantity is to be detected. To analyze this input, we shall apply an ANN. This will serve as our benchmark in Section 6 where we shall use the ANN together with a preknowledge of our physical system to synthesize a virtual training set for out-of-reach scenarios.

3.1. Study Case I: Cantilever Beam Deflection Detection. In some cases, the frequency response of a loaded cantilever beam is examined to detect the load magnitude (as in [43] where femtograms are detected) and position [44]. Obviously, these cases use beams with known dimensions. Consider a case where the loads (magnitudes and positions) are to be detected on a beam which is already in use. So, the beam's cross-section dimensions are not necessarily known for reasons mentioned above. Such a case can emerge if (1) internal cavities within the beam may exist, (2) a time-dependent wear is significant, and (3) cross-section measurements cannot be performed for any reason.

In this study, in order to find the loads' magnitudes and positions along a cantilever beam, a set of strain gauges are used with the wiring depicted in Figure 5(a). If the beam's cross-section is known, the strain field along the beam can provide the magnitude and position of a load. The expected stress field in this case would nullify from the free end to the load position. The force magnitude can be extracted using Euler-Bernoulli beam equation $\varepsilon(x) = M(x)y(x)/I(x)E$. Nevertheless, in the case where a second moment of area $I(x)$ and the position from the neutral axis $y(x)$ are unknown, this would not do. Therefore, one can conduct a set of experiments with a single load at a time; these would constitute a training set for an ANN that would be able to extract the load properties from the strain field. Figure 1 presents a cantilever beam with a nonuniform cross-section. The set of strain gauges is fixed along the beam axis on its bottom face. For

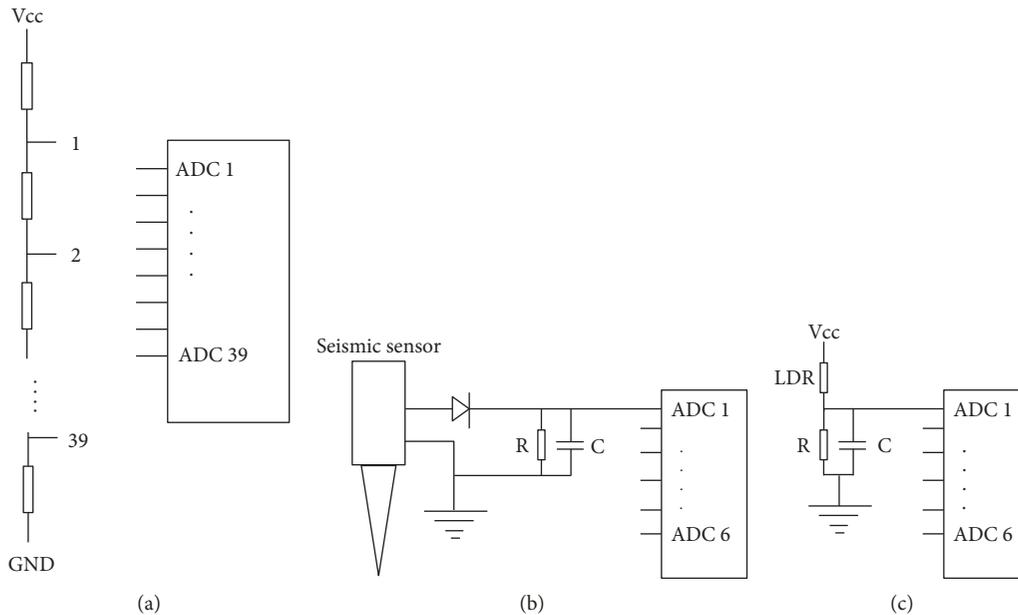


FIGURE 5: Schematic drawings of the sensor arrays: (a) a serial connection wiring design of strain gauges and (b) the connections of the seismic sensor where we used a diode to filter the positive voltage alone and a RC circuit for measuring the maximal amplitude during the impulse response. The nodes are connected to the microcontroller. (c) The LDR wiring where the light intensity was controlled by a PWM signal and a capacitor was used as a LPF.

simplicity, the loads that shall be considered are uniformly distributed as depicted to avoid torsion effect. Along the experiment, distributed forces of magnitude 5–15 kg were loaded one at a time on a 1 m long cantilever beam. At the second stage of this paper, we load the beam with two simultaneous forces of the same range of magnitudes.

3.2. Study Case II: Metal Sheet Impulse Detection. Solids support sound vibrations of different types. Seismic sensors are devices used to measure these vibrations by converting motion into a measurable electronic signal. Several attempts to develop an automatic means to detect and locate mechanical impacts are available in literature (see for example, [45] where inspections of metallic-based structures are proposed and in [46], a sensor system is devised to monitor impact damage of pressure vessels). The advantage of using an ANN for this application is realized for reasons mentioned above.

In this study, we attached six *geophone* seismic sensors to a $380 \times 440 \times 2$ mm aluminum plate. The sensor's positions were selected arbitrarily and the wiring is depicted in Figure 5(b). A 75 mm long and 19 mm diameter brass rod was dropped on the plate from a 20 mm height. At each drop, the sensor data was collected using a microcontroller. For the sake of accuracy and repeatability, we used a *ScorBot* serial 5 DOF robot to drop the brass rod on the plate in a 5 mm resolution grid distribution. We repeated the experiment at the second stage of this paper where we dropped the same rod from a 40 mm height a few additional times in order to interpolate synthetic experiments (i.e., to extract data from non-existent experiments performed from varying heights).

3.3. Study Case III: Light Sensor Array. In order to demonstrate a nontrivial f , we devised a linear sensor array as a low-cost *Lidar*. The LEDs' positions were not carefully aligned with the LDRs; the wiring is depicted in Figure 5(c). A similar optical light array concept is used in [19]; in their work, they used TSL1401 Linear Array of photodiodes which is commonly used for edge detection for printers. The researchers used two commercially available IR line image sensors and a 128×1 array photo detector to measure the position of the finger and its gesture. For simplicity, here, we used LDRs.

4. ANN Architecture and Training

This section presents the ANNs and their training setups used for the study cases. For all cases, we used MATLAB's neural network fitting as a generator with a two-layer feed-forward network with Sigmoid hidden perceptrons and linear output perceptrons having 100 perceptrons in the hidden layer. The network was trained with Levenberg-Marquardt backpropagation algorithm. 70% of the experimental data was used for training, 15% for validation, and 15% for testing.

In study case I, the neural network is given a set of 50,000 experiments each having a vector of length 39 capturing the stress field under each loading. The output is a vector of length 2 representing the load's position and magnitude. We used a finite element method to calculate the stress field and used mean square error of the output vector as the cost (error) function. We consider a uniform steel cantilever beam with a varying cross-section. The beam model is

developed with the FEM Toolbox for MATLAB [47]; using 40×16 elements, we set Poisson's ratio ν to be 0.3 and $\rho = 7,890 \text{ kg/m}^3$.

In study case II, a set of 1,925 experiments each is a vector of size 6 capturing the geophone magnitudes. A vector of length 2 representing the impulse xy position on the plate constituted the output.

In study case III, the neural network is given a set of 50,000 real experiments each having a vector of length 6 capturing the voltage under each imminence. The output is a vector of length 6 representing the estimated magnitudes at the six possible positions of the six LEDs.

5. Results for the Single-Quantity Cases

In study case I, each FEM run took about 2 sec. Experimental data from 50,000 scenarios were collected (two days is a reasonable estimation for the time required for a real robotic experiment assuming ~ 3 sec for each load). The training phase took about 10 minutes. Figure 6 plots the model error histogram across 50,000 unseen experiments. The average position error and the standard deviation extracted using A_1 were $\mu = 6 \text{ mm}$ and $\sigma = 3.7 \text{ mm}$, respectively. The average load error and the standard deviation extracted were $\mu = 1 \cdot 10^{-2} \text{ gr}$ and $\sigma = 1 \cdot 10^{-2} \text{ gr}$, respectively. In study case II, the data collection process took about 3.5 hours during which experimental data from 1,925 plate impulses in different positions were collected. Figure 7 plots the model error histogram across 131 unseen experiments. The average drop position error and standard deviation using A_1 were $\mu = 11 \text{ mm}$ and $\sigma = 5.8 \text{ mm}$, respectively. In study case III, the data collection process took about 2 hours during which data from 50,000 real experiments was collected. At each experiment, a single random LED was turned on with a random duty cycle. Figure 8(a) plots the magnitude error histogram across 20,000 unseen experiments. The histogram is given as a function of percentile error of the LED activation duty cycle (which corresponds to the light intensity one senses). In $\sim 85\%$ of the cases, A_1 succeeded in assessing the true position of the light source. The duty cycle error average and standard deviation were $\mu = 13.4\%$ and $\sigma = 19\%$, respectively.

6. Bootstrapping Data Sets and ANN Training for the Multiple-Quantity Cases

In all cases, we used the same ANN architecture as described in Section 4.

The cantilever beam sensor in study case I is capable of locating multiple loads as well. For this end, the training task requires multiple simultaneously independent loadings. Nevertheless, recall that an ANN requires a large data set, for example, two loads would require a data set of magnitude $\approx 50,000^2$ to have the accuracy described in Section 5. This obviously is out of reach if this should be mechanically accumulated (i.e., actually loading the beam).

To overcome this, we bootstrap our single-load training data to that of multiple loads. In other words, we construct synthetic data from our already established one-load train-

ing data to train an additional ANN, which we marked A_3 in Figure 4, for multiple loads. Since we assume our loads do not exceed the elastic region, we may use superposition which corresponds to map H of Figure 4. In other words, the resultant strain caused by several loads applied simultaneously is the summation of the strain caused by each load individually.

Figure 9 presents the results of synthetic training A_3 for two loads. The average position error and the standard deviation extracted using A_3 were $\mu = 50 \text{ mm}$ and $\sigma = 38 \text{ mm}$, respectively. The average load error and the standard deviation extracted using A_3 were $\mu = 0.16 \text{ gr}$ and $\sigma = 0.14 \text{ gr}$, respectively.

Study case II demonstrates our method for the case of finding the position of an unknown impulse magnitude (we do so since for study case II, simultaneous impulses will require a nonrealistic time proximity.). Recall that in Section 4, we dropped the brass rod from a constant height of 20 mm. Locating the impulse position for different impulse magnitudes requires data training for these height drops. Nevertheless, since we use a capacitor, an increase in the impulse magnitude will result in an increased output voltage of the sensor. Knowing the sensor's output resulting from two different impulses located at the same position, one may use a simplified linear description (i.e., linear interpolation) to calculate the sensor's output for an impulse in that range. To this end, we conduct 72 additional high-altitude drops (i.e., of large magnitude) to generate the synthetic data for ANN A_3 (see Figure 4). Explicitly, we define $4 \times 6 = 24$ rectangular regions that cover the plate. For each region, we conducted an additional 3 drops at its center. These were then used to give 24 linear approximations, corresponding to map H of Figure 4, for sensor data that corresponds to impulses of a wider magnitude range. Based on these experiments, we generated 194,000 synthetic data records that correspond to drops of a wide magnitude range. These were then used to train ANN A_3 . Figure 10(b) presents the results. The average position error and the standard deviation extracted using A_3 were $\mu = 15.2 \text{ mm}$ and $\sigma = 12.8 \text{ mm}$, respectively.

Remark. In order to demonstrate the efficiency of our method, we examined A_1 which was trained for only 20 mm height drops as in Section 5. The ANN was tested with experimental data from varying height drops. Figure 10(a) presents the results—an average position error and the standard deviation extracted were $\mu = 52.7 \text{ mm}$ and $\sigma = 55.4 \text{ mm}$, respectively.

The sensor in study case III is capable of locating multiple light sources as well. For this end, the training task requires multiple simultaneous luminescence. An experiment set for a single light source theoretically requires 6×255 experiments which amount to only ~ 7 minutes of data harvesting. But note that since we use here real experiments as training data, errors and inaccuracies are inevitable, so a limited data set did not suffice and we were forced to use 50,000 experiments to train the net. The experiment took ~ 2 hours to complete.

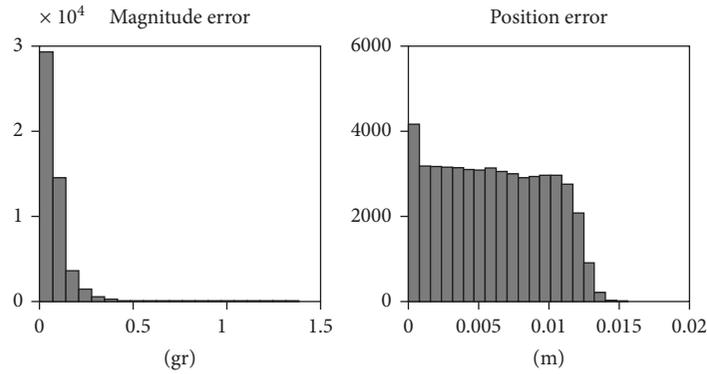


FIGURE 6: Study case I: the resulting ANN error histogram (of the magnitude and position) of a single load applied on the cantilever beam.

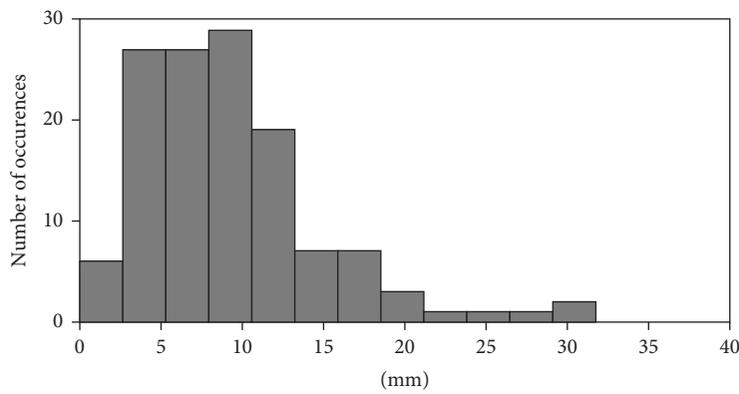


FIGURE 7: Study case II: the error histogram of a single impulse applied on the plate. Although the dropping training set was of a 5 mm resolution, the resulting accuracies are 2 – 3 mm bigger. This can be attributed to the repetitively small training data set and the errors in the magnitudes of the impulses.

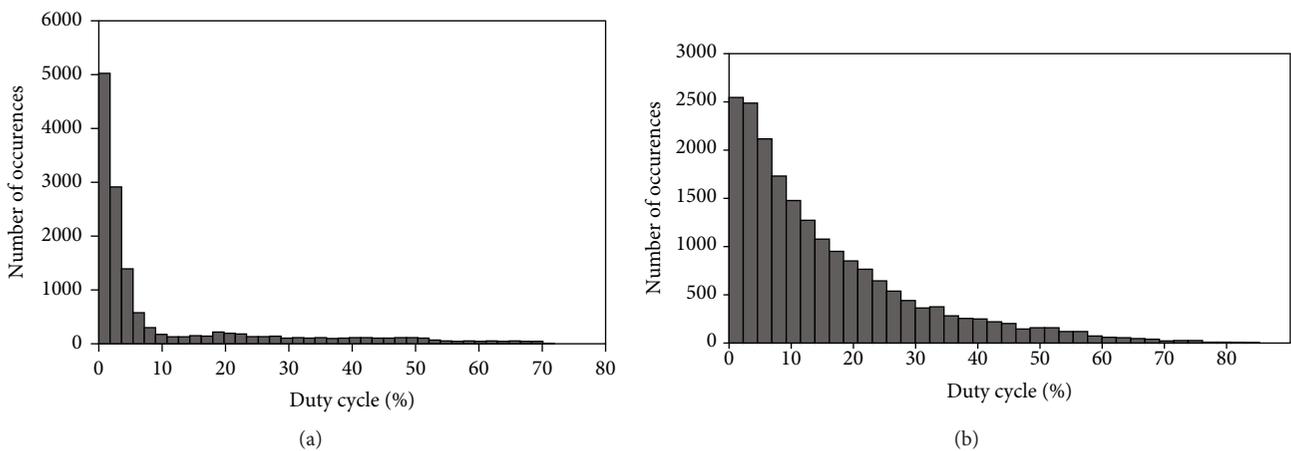


FIGURE 8: Study case III: the resulting ANN error histograms. The x -axis represents the estimation error in duty cycle percentage. (a) Histogram for 20,000 unseen experiments for a single light source and (b) histogram for 20,000 unseen experiments for a pair of simultaneous light sources where the x -axis represents the estimation error of the second light source while the first was accurately estimated.

Obviously, when two light sources are of interest, a much larger number of experiments are required. To overcome this, we used equation (2) to bootstrap our single light source training data to that of two lights. The bootstrapping procedure for generating 100,000 synthetic data “experiments”

took 2 sec. This was used to train the ANN, which we marked A_3 in Figure 4, for multiple lights.

Figure 8(b) presents the results of the synthetic training of A_3 for the two light source cases across 20,000 unseen experiments. The average intensity error and the standard

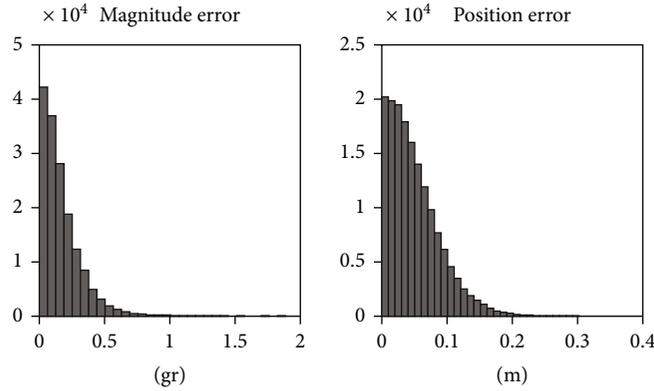


FIGURE 9: Study case I: the absolute error histogram of two loads applied on the beam.

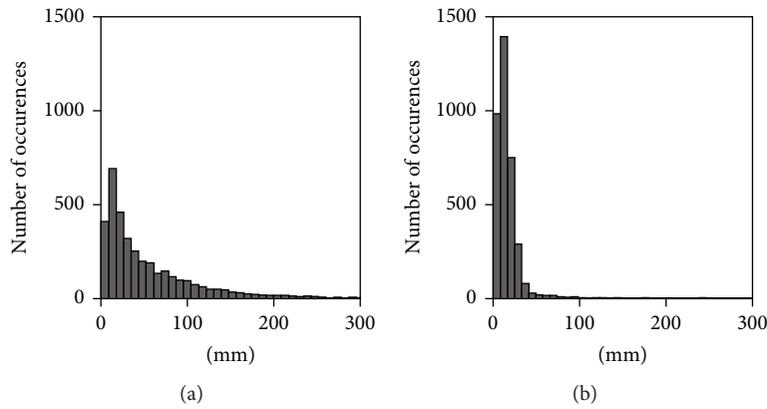


FIGURE 10: Study case II: (a) the A_1 ANN ability to calculate the positions of random magnitude drops. (b) The performance of ANN A_3 for estimating the positions of random magnitude drops. A_1 was trained with 1,950 constant magnitude impulses collected from real experiments. A_3 was trained with 194,000 synthetic data records which were based on additional 72 real experiments.

deviation were $\mu = 14.7$ and $\sigma = 21.5$, respectively (given in duty cycle percentage). The ANN provided a $\sim 70\%$ success in accurately positioning two light sources with an additional $\sim 20\%$ of the tests which showed an accurate estimation of one of the light source positions while the other was estimated neighboring the true position.

Remark. Note that the discussion above assumes that the number of loads/light sources is known. This is a rather easy task for an ANN. For example, to establish this for case I, one may use the simplest classifying ANN with a one-layer feedforward 50-perceptron architecture. We did so by training the ANN with the following data: (i) 25,000 acquired data entries from single-load experiments, with additional (ii) 25,000 synthetic data entries which simulate experiments with two loads. The ANN succeeded in categorizing 84% of the cases. Of course, once one knows how many loads participate in a given experiment, the ANNs described in the paper can be applied.

7. Conclusions

We have presented three sensor array scenarios. In Section 4, we used ANN to identify the nonlinear function between the sensor's output voltages and the measured quantity (a static load, an impulse, and a light source detection). For the first

and third sensor array cases, we demonstrated how one can bootstrap a synthetic data set of multiple stimuli from a data set of a single stimulus. For the second case, we showed how synthetic data for unknown impact magnitudes can be bootstrapped from a data set of an impact of a constant magnitude. Such a strategy is essential when a data set which should be extracted from a real experimental setup is not feasible.

Our results show that generating synthetic data while carefully considering the physical environment is efficient both in the sense of time resources and accuracy. For cases II and III for example, the time required for producing a training set for A_3 was about two orders of magnitude shorter.

In this paper, we used very simple and shallow ANNs A_1 and A_3 . Obviously, using more sophisticated architectures may improve the estimation accuracy if needed. A previous research work by the authors attempted to design a spatial shape sensor for flexible manifolds. Future work will apply the method presented here together with a deep-learning ANN for estimating spatial shapes in real time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] W. Cheng, L.-L. Zhao, and C.-L. Fu, "Source term identification for an axisymmetric inverse heat conduction problem," *Computers & Mathematics with Applications*, vol. 59, no. 1, pp. 142–148, 2010.
- [2] M. Mierzwiczak and J. A. Kołodziej, "The determination of heat sources in two dimensional inverse steady heat problems by means of the method of fundamental solutions," *Inverse Problems in Science and Engineering*, vol. 19, no. 6, pp. 777–792, 2011.
- [3] A. Erdem, D. Lesnic, and A. Hasanov, "Identification of a spacewise dependent heat source," *Applied Mathematical Modelling*, vol. 37, no. 24, pp. 10231–10244, 2013.
- [4] K. Rashedi, H. Adibi, and M. Dehghan, "Determination of space-time-dependent heat source in a parabolic inverse problem via the Ritz–Galerkin technique," *Inverse Problems in Science and Engineering*, vol. 22, no. 7, pp. 1077–1108, 2014.
- [5] A. Frąckowiak, N. D. Botkin, M. Ciałkowski, and K.-H. Hoffmann, "Iterative algorithm for solving the inverse heat conduction problems with the unknown source function," *Inverse Problems in Science and Engineering*, vol. 23, no. 6, pp. 1056–1071, 2015.
- [6] O. Medina, A. Shapiro, and N. Shvalb, "Resistor-based shape sensor for a spatial flexible manifold," *IEEE Sensors Journal*, vol. 17, no. 1, pp. 46–50, 2017.
- [7] P. Ghaderi, A. J. Dick, J. R. Foley, and G. Falbo, "Practical high-fidelity frequency-domain force and location identification," *Computers & Structures*, vol. 158, pp. 30–41, 2015.
- [8] J. Liu, X. Sun, X. Han, C. Jiang, and D. Yu, "A novel computational inverse technique for load identification using the shape function method of moving least square fitting," *Computers & Structures*, vol. 144, pp. 127–137, 2014.
- [9] J. Liu, X. Sun, X. Han, C. Jiang, and D. Yu, "Dynamic load identification for stochastic structures based on Gegenbauer polynomial approximation and regularization method," *Mechanical Systems and Signal Processing*, vol. 56–57, pp. 35–54, 2015.
- [10] D. Lesnic and A. Zeb, "The method of fundamental solutions for an inverse internal boundary value problem for the biharmonic equation," *International Journal of Computational Methods*, vol. 6, no. 4, pp. 557–567, 2009.
- [11] S. Chesne, C. Pezerat, and J.-L. Guyader, "Identification of plate boundary forces from measured displacements," *Journal of Vibration and Acoustics*, vol. 130, no. 4, article 041006, 2008.
- [12] C. Pezerat and J. L. Guyader, "Identification of vibration sources," *Applied Acoustics*, vol. 61, no. 3, pp. 309–324, 2000.
- [13] X. Q. Jiang and H. Y. Hu, "Reconstruction of distributed dynamic loads on a thin plate via mode-selection and consistent spatial expression," *Journal of Sound and Vibration*, vol. 323, no. 3–5, pp. 626–644, 2009.
- [14] M. Chierichetti and M. Ruzzene, "Dynamic displacement field reconstruction through a limited set of measurements: application to plates," *Journal of Sound and Vibration*, vol. 331, no. 21, pp. 4713–4728, 2012.
- [15] A. Berry, O. Robin, and F. Pierron, "Identification of dynamic loading on a bending plate using the virtual fields method," *Journal of Sound and Vibration*, vol. 333, no. 26, pp. 7151–7164, 2014.
- [16] K. Li, J. Liu, X. Han, X. Sun, and C. Jiang, "A novel approach for distributed dynamic load reconstruction by space-time domain decoupling," *Journal of Sound and Vibration*, vol. 348, pp. 137–148, 2015.
- [17] Y. Zhang and J. A. Mann III, "Measuring the structural intensity and force distribution in plates," *The Journal of the Acoustical Society of America*, vol. 99, no. 1, pp. 345–353, 1996.
- [18] J. Sanchez and H. Benaroya, "Review of force reconstruction techniques," *Journal of Sound and Vibration*, vol. 333, no. 14, pp. 2999–3018, 2014.
- [19] S.-C. Lim, J. Shin, S.-C. Kim, and J. Park, "Expansion of smart-watch touch interface from touchscreen to around device interface using infrared line image sensors," *Sensors*, vol. 15, no. 7, pp. 16642–16653, 2015.
- [20] H. Everett, *Sensors for Mobile Robots: Theory and Application*, AK Peters Ltd., 1995.
- [21] W. Sansen, J. Huijsing, and R. J. van de Plassche, *Analog Circuit Design: Mixed A/D Circuit Design Sensor Interface Circuits and Communication Circuits*, Springer Science & Business Media, 2013.
- [22] O. Medina, A. Shapiro, and N. Shvalb, "Kinematics for an actuated flexible n -manifold," *Journal of Mechanisms and Robotics*, vol. 8, no. 2, article 021009, 2016.
- [23] E. Hristoforou and R. E. Reilly, "Tensile stress distribution sensors based on amorphous alloys," *Journal of Magnetism and Magnetic Materials*, vol. 119, no. 3, pp. 247–253, 1993.
- [24] X. Roset, J. Del Rio, A. Manuel, and R. Palomera-Garcia, "Contributions to model characterization of geophone sensor," in *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.04CH37510)*, pp. 1896–1899, Como, Italy, 2004.
- [25] T. Lozano-Perez, I. J. Cox, and G. T. Wilfong, *Autonomous Robot Vehicles*, Springer Science & Business Media, 2012.
- [26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, Published by MIT press, 2012.
- [27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [28] R. Vicen-Bueno, M. Rosa-Zurera, M. P. Jarabo-Amores, and D. de la Mata-Moya, "Coherent detection of Swerling 0 targets in sea-ice Weibull-distributed clutter using neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3139–3151, 2010.
- [29] J. Huang, H. Yuan, Y. Cui, and Z. Zheng, "Nonintrusive pressure measurement with capacitance method based on FLANN," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 11, pp. 2914–2920, 2010.
- [30] A. Malhi, R. Yan, and R. X. Gao, "Prognosis of defect propagation based on recurrent neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703–711, 2011.
- [31] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Computer Vision – ECCV 2012. ECCV 2012*, pp. 611–625, Springer, 2012.

- [32] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: ground truth from computer games," in *Computer Vision – ECCV 2016*, pp. 102–118, Springer, 2016.
- [33] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4077–4085, Las Vegas, NV, USA, 2016.
- [34] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, Vancouver, BC, Canada, 2017.
- [35] F. Sadeghi and S. Levine, "Cad2rl: real single-image flight without a single real image," 2016, <http://arxiv.org/abs/1611.04201>.
- [36] A. Dosovitskiy, P. Fischer, E. Ilg et al., "FlowNet: learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766, Santiago, Chile, 2015.
- [37] N. Mayer, E. Ilg, P. Hausser et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4040–4048, Las Vegas, NV, USA, 2016.
- [38] R. McHaney, *Computer Simulation: A Practical Perspective*, Academic press, 1991.
- [39] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 49–56, San Francisco, CA, USA, 2010.
- [40] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, CRC press, 1994.
- [41] O. Medina and R. Yozevitch, "A neural network based omni light sensor," in *2017 13th IEEE International Conference on Control & Automation (ICCA)*, pp. 872–875, Ohrid, Macedonia, 2017.
- [42] F. Neuman, "Finite sums of products of functions in single variables," *Linear Algebra and its Applications*, vol. 134, pp. 153–164, 1990.
- [43] J. Teva, G. Abadal, F. Torres, J. Verd, F. Perez-Murano, and N. Barniol, "A femtogram resolution mass sensor platform based on SOI electrostatically driven resonant cantilever. Part II: sensor calibration and glycerine evaporation rate measurement," *Ultramicroscopy*, vol. 106, no. 8-9, pp. 808–814, 2006.
- [44] S. Dohn, W. Svendsen, A. Boisen, and O. Hansen, "Mass and position determination of attached particles on cantilever based mass sensors," *Review of Scientific Instruments*, vol. 78, no. 10, article 103303, 2007.
- [45] C. Morón, M. Portilla, J. Somolinos, and R. Morales, "Low-cost impact detection and location for automated inspections of 3D metallic based structures," *Sensors*, vol. 15, no. 6, pp. 12651–12667, 2015.
- [46] S. Atobe, S. Sugimoto, N. Hu, and H. Fukunaga, "Impact damage monitoring of FRP pressure vessels based on impact force identification," *Advanced Composite Materials*, vol. 23, no. 5-6, pp. 491–505, 2014.
- [47] Z. Anton, "MATLAB FEM Toolbox for solid mechanics," <http://www.codedevelopment.net>.

