sensors

Article

Machine Learning for Touch Localization on an Ultrasonic Lamb Wave Touchscreen

Sahar Bahrami [1], Jérémy Moriot [1,2] , Patrice Masson [1] and François Grondin [1,*]

[1] Faculty of Engineering, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada; sahar.bahrami@usherbrooke.ca (S.B.); j.moriot@allwavestechnologies.com (J.M.); patrice.masson@usherbrooke.ca (P.M.)
[2] All Waves Technologies, Sherbrooke, QC J1N 0C8, Canada
* Correspondence: francois.grondin2@usherbrooke.ca

Abstract: Classification and regression employing a simple Deep Neural Network (DNN) are investigated to perform touch localization on a tactile surface using ultrasonic guided waves. A robotic finger first simulates the touch action and captures the data to train a model. The model is then validated with data from experiments conducted with human fingers. The localization root mean square errors (RMSE) in time and frequency domains are presented. The proposed method provides satisfactory localization results for most human–machine interactions, with a mean error of 0.47 cm and standard deviation of 0.18 cm and a computing time of 0.44 ms. The classification approach is also adapted to identify touches on an access control keypad layout, which leads to an accuracy of 97% with a computing time of 0.28 ms. These results demonstrate that DNN-based methods are a viable alternative to signal processing-based approaches for accurate and robust touch localization using ultrasonic guided waves.

Keywords: signal processing; machine earning; ultrasonic wave; touch technology; human–machine interface

check for updates

Citation: Bahrami, S.; Moriot, J.; Masson, P.; Grondin, F. Machine Learning for Touch Localization on an Ultrasonic Lamb Wave Touchscreen. *Sensors* **2022**, *22*, 3183. https://doi.org/10.3390/s22093183

Academic Editor: Christoph M. Friedrich

Received: 8 March 2022
Accepted: 18 April 2022
Published: 21 April 2022

1. Introduction

The popularity of tactile sensor technologies in daily life (e.g., cell phones, access keypads, smart screens) leads to an increase in the demand for low-cost robust touch interfaces. Amongst the existing technologies [1–3], tactile surfaces based on ultrasonic guided waves answer the need for converting non-planar surfaces (including plastic material and transparent glass) into high resolution and durable tactile sensing surfaces. Moreover, this technology supports multi-touch detection and can estimate the contact pressure [4,5]. Acoustic based tactile sensors exploit the Surface Acoustic Waves (SAW) [6,7], or guided or Lamb Waves (LW) [4,8–13]. SAWs travel over the surface [14], whereas LWs propagate throughout the thickness of the material. The SAW tactile surfaces are vulnerable to surface contaminants such as liquid and scratches, and can only offer limited multiple-finger commands. On the other hand, Lamb Wave Touchscreen (LWT) technology supports multi-touch, which makes it more suitable for interfacing with smart devices [4,5]. LWT technologies are either passive [8,11,15–17] or active [4,9,10,12,13,18–21]. In passive LWT, touch action is the source of Lamb waves which can be identified by piezoceramic sensors located on the plate. In active LWT, the Lamb waves are both generated and received by piezoceramic elements. Static actions are not taken into account in passive mode since they do not create the wave field, while the active mode can overcome this deficiency [5,21].

While LWT offers interesting properties for numerous applications, it remains challenging to perform accurate touch localization. Many localization techniques have been proposed to address this challenge [22], some of them better suited to address the additional challenge of designing low-cost embedded systems. Some imaging-based algorithms localize the tactile position using Time of Flight (ToF) [23–26] and Delay-and-Sum beamforming [27]. Other techniques utilize damage indices (DIs) [28], which rely on correlation coefficients [29,30] to measure the similarity between the baseline and the input signals. Alternatively, localization can also be achieved using data-based learning methods [13,20]. Quaegebeur et al. [4,19] also proposed an ultrasonic touch screen based on guided wave interaction with a contact impedance. This technology is used to implement the glass tactile surface in this work and supports a wide range of screen sizes and solid materials. In this configuration, piezoceramic discs are used as four receivers and one emitter, installed around the plate. The contact impedance induces reflections and modifies the ultrasonic guided wave field. A Generalized Cross-Correlation (GCC) algorithm computes the similarity between the measured signal and a set of recorded signals to estimate the touch position. However, this approach requires costly hardware to process the large dictionary of reference signals, as it involves a significant amount of memory access and floating point operations (flops).

Besides this, Artificial Neural Networks (ANNs) used in Artificial Intelligence (AI) are applied in Structural Health Monitoring (SHM) [31,32] for damage identification and localization. A Multi-Layer Perceptron (MLP) is implemented for damage localization and quantification using the damage index. The k-Nearest Neighbor (kNN) algorithm with a Principal Component Analysis (PCA) feature extraction method is also investigated in [33]. A Support Vector Machine (SVM) for damage classification and localization was studied in [34], with features from different domains (time and frequency). The accuracy of this method, however, remains sensitive to noise and degrades to 57% when the Signal-to-Noise Ratio (SNR) drops by 30 dB. An auto-encoder-based approach for acoustic emission sources localization is studied in [35]. The RMS localization errors are 38 mm and 48 mm (for two metallic panels), which represents an improvement in comparison with SVM and ANN (78 mm and 67 mm, respectively). A singular value decomposition (SVD-PHAT) was investigated in [36,37] to localize multiple sound sources. The proposed method offers localization performance similar to SRP-PHAT but considerably reduces the computational load. Deep learning-based approaches have also been applied in SHM. Many SHM studies exploit Convolution Neural Network (CNN)-based models [38–44]. Rautela et al. [38] obtained damage detection and localization accuracy of 99% using 1D CNN. In the signal pre-processing phase, they performed five operations: (1) band-pass filtering and visualization; (2) frequency preferencing; (3) signal augmentation with noise; (4) cross-statistical feature engineering; and (5) TOF.

Other approaches from AI, such as Machine learning (ML) and Deep learning (DL) methods, can determine the coordinates of a finger in contact with a surface. A study has demonstrated that CNN can identify left and right thumbs with an accuracy of over 92% on capacitive touchscreens [45]. Chang and Lee [21] proposed a deep machine learning algorithm of 2D CNN for Lamb wave localization on an ultrasonic touch screen using 48 actuator-sensor paths. The 16 piezoceramic transducers contain 4 transmitters and 12 receivers. The CNN input data are an image with the dimensions of 12 × 2048 pixels, including data points gathered by the receivers, and achieved a positioning accuracy of 95%. To improve the localization performance, Li [22] proposed an ML-based lamb waves scatterer localization method, called CNN-GCCA-SVR. They trained a CNN-GCCA (deep version of generalized canonical correlation analysis) to extract the features, then used a Support Vector Regression (SVR) model for the localization task. This algorithm provides precise prediction using only one actuator and two sensors, with the localization errors between 2 mm and 12 mm depending on the sensing configurations.

The development of ML and DL methods in relation with ultrasonic Lamb wave technologies in recent years, and the demonstrated performance of AI in localization as mentioned above, encourage further investigations on ML and DL algorithms to improve accuracy in touch localization. In this study, a simple Deep Neural Network (DNN) is proposed to localize a finger in contact with a surface, using classification and regression approaches. The proposed approach also shows that a neural network can be trained for a specific keyboard layout to perform classification with high accuracy. To the best of our knowledge, this is the first time that a simple fully connected neural network is used for a robust touch localization on a tactile surface when exploiting ultrasonic guided waves. This work is organized as follows: in Section 2, the hardware setup is described. The dataset is introduced in Section 3. In Section 4, the localization methods employing classification and regression are investigated. The results are discussed in Section 4.3. The applications of the methods including access control keypad (Section 5.1) and tracking touch by human fingers (Section 5.2) are introduced in Section 5, which is followed by a conclusion (Section 6).

2. Hardware Setup

Figure 1 shows the processing pipeline. An array of five piezoceramic elements soldered to a flexible Printed Circuit Board (PCB) is bonded to the touch surface with regular epoxy. A custom LabVIEW interface from National Instruments (NI) is used to (1) control an NI-9262 module that emits an ultrasound signal through a linear amplifier driving the emitting piezoceramic, (2) control an NI-9223 acquisition module to record ultrasound signals measured by the four receiving piezoceramics amplified by a custom preamplifier, and (3) control a collaborative Universal Robot UR5e having six degrees of freedom (DoFs), which touches a glass surface with an artificial silicon finger at a desired position.

During the acquisition process, an emission signal excites the emitting piezoceramic element, which converts it into a mechanical wave that propagates through the host structure. When a finger touches the structure, it modifies the surface mechanical impedance, inducing reflections of the mechanical waves. These reflected waves are measured by the receiving piezoceramics, which convert them into electrical signals.

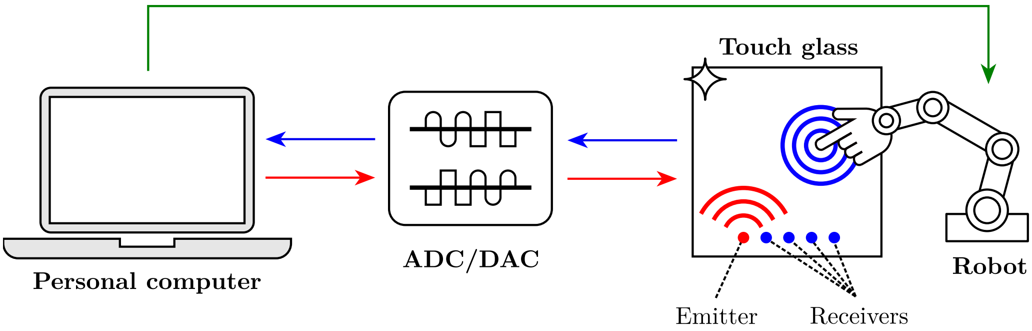

Figure 1. Schematic of the experimental setup.

The touch surface is a tempered glass plate (shown in Figure 2) with dimensions of 20 × 20 cm and a thickness of 5 mm. The four receivers and the emitter are installed at the bottom of the glass, and consist of piezoceramic discs with a diameter of 6 mm and a thickness of 2 mm. The Universal Robot UR5e is equipped with a silicon finger with a diameter of 5 mm. Silicon is a material of choice for this application, as its mechanical impedance is close to a human finger. The x and y coordinates of the contact, such as contact pressure, are controlled accurately by the UR5e and recorded by the computer, which provides a reliable baseline.

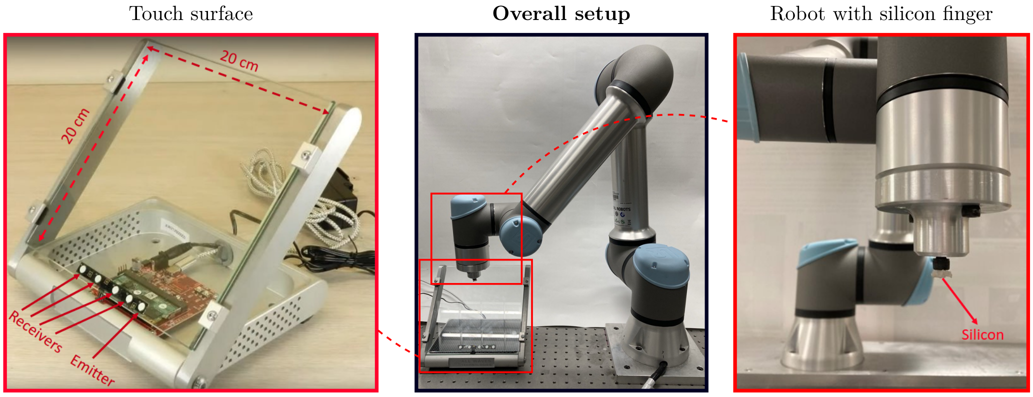

Figure 2. Hardware setup with the touch glass and the robotic arm.

3. Dataset

The robotic arm produces 6404 contacts at random positions and pressure on the glass surface. We set the robot to acquire random contact positions over the touch glass. The experiment was done for a given amount of time until a sufficient amount of data is generated. This dataset is split into training (70%), validation (20%), and test set (10%) signals. For each touch, a linear chirp excitation signal of 1 ms duration from 50 kHz to 100 kHz is generated with a sampling frequency of 500 kHz. The receiver signals are acquired during 1 ms at a sample rate of 250 kHz. Figure 3 shows how the features are extracted in the time and frequency domains. In the time domain, the 250-sample signals acquired for each of the four receivers are simply concatenated. In the frequency domain, the real and imaginary parts in the frequency range between 50 kHz and 100 kHz are selected from the Fast Fourier Transform (FFT) after applying a Hann window, which leads to four vectors of 49 elements. These vectors are concatenated to create a feature vector of 392 elements.

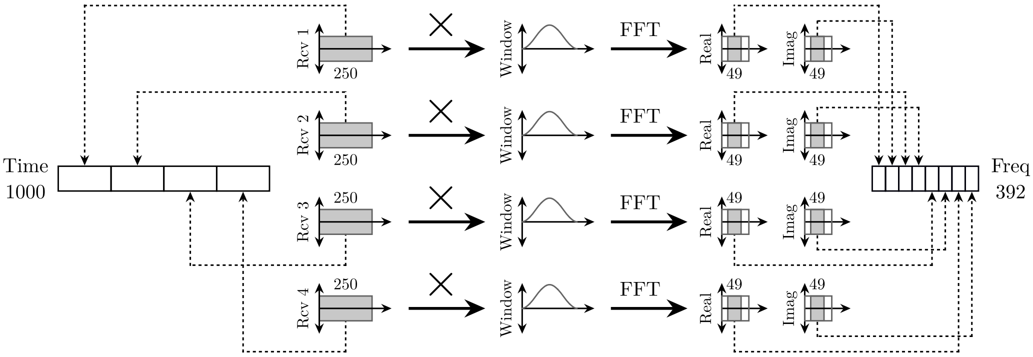

Figure 3. Features extraction in the time and frequency domains.

4. Localization

Localization consists of predicting the horizontal and vertical coordinates of the finger on the touch surface. The model takes the input signal in the time or frequency domain, and predicts the 2D-coordinate with a discrete grid using a classification approach, or in a continuous space using a regression approach.

4.1. Classification Approach

A classification approach is first investigated to estimate the contact point location on the touch surface. The 20×20 cm^2 surface is first divided into $N \times N$ zones (or classes) of sizes $\frac{20}{N} \times \frac{20}{N}$ cm^2, as shown in Figure 4. When class (i, j) is activated (where $i \in \{1, \dots, N\}$ stands for the row index and $j \in \{1, \dots, N\}$ for the column index), the estimated position of the touch contact corresponds to the center of mass of the zone, denoted as $\mathbf{c}_{i,j}$.

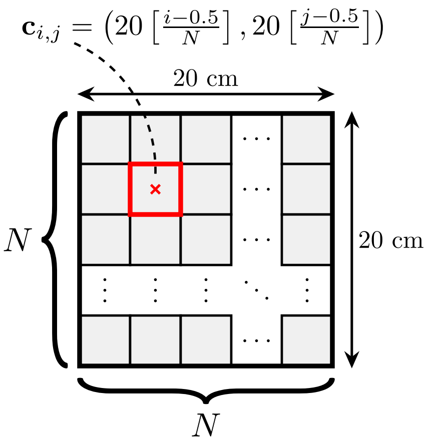
$\mathbf{c}_{i,j} = \left(20\left[\frac{i-0.5}{N}\right], 20\left[\frac{j-0.5}{N}\right]\right)$
20 cm
20 cm
N
N

Figure 4. Classification touch zones. When zone (i, j) gets selected, the estimated touch position corresponds to the center of mass of this zone, denoted as $\mathbf{c}_{i,j}$.

Table 1 shows the nine different configurations explored in this work. As the grid resolution increases, the center of mass of each class gets closer to the exact touch position. However, the classification task also becomes more challenging, which can reduce the localization accuracy when the wrong class is selected.

Table 1. Configurations used for touch zone classification.

Grid Resolution (N)	Number of Classes (N^2)	Class Area (cm^2)
2	4	10.0
3	9	4.4
4	16	2.5
5	25	1.6
6	36	1.1
7	49	0.8
8	64	0.6
9	81	0.5
10	100	0.4

Figure 5 shows a four-stage fully connected neural network to perform classification based on the input vector in the time ($\mathbf{x} \in \mathbb{R}^{1000}$) or frequency ($\mathbf{x} \in \mathbb{R}^{392}$) domain. Each stage consists of a cascade of a batch normalization layer, a linear layer, and a rectified linear unit activation function. The first stage outputs the tensor $\mathbf{h}_1 \in \mathbb{R}^{400}$, while the second, third, and fourth stages generate $\mathbf{h}_2 \in \mathbb{R}^{300}$, $\mathbf{h}_3 \in \mathbb{R}^{200}$ and $\mathbf{h}_4 \in \mathbb{R}^{100}$, respectively. A training time, a dropout layer where neurons are zeroed with a probability of 0.3 ensures regularization and prevents overfitting. Finally, the classification stage then consists of a linear layer followed by a softmax layer, which produces a one-hot vector of $\hat{\mathbf{y}} \in [0,1]^{N^2}$ elements, with all zone indices concatenated ($\hat{\mathbf{y}} = \left[\hat{y}_{(1,1)}, \hat{y}_{(1,2)}, \ldots, \hat{y}_{(i,j)}, \ldots, \hat{y}_{(N,N)}\right]$).

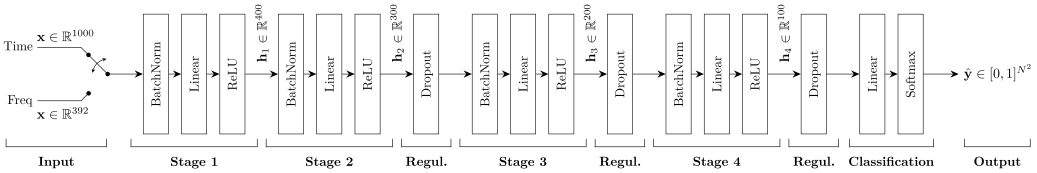
Time
$\mathbf{x} \in \mathbb{R}^{1000}$
Freq
$\mathbf{x} \in \mathbb{R}^{392}$
BatchNorm
Linear
ReLU
$\mathbf{h}_1 \in \mathbb{R}^{400}$
BatchNorm
Linear
ReLU
$\mathbf{h}_2 \in \mathbb{R}^{300}$
Dropout
BatchNorm
Linear
ReLU
$\mathbf{h}_3 \in \mathbb{R}^{200}$
Dropout
BatchNorm
Linear
ReLU
$\mathbf{h}_4 \in \mathbb{R}^{100}$
Dropout
Linear
Softmax
$\hat{\mathbf{y}} \in [0,1]^{N^2}$
Input
Stage 1
Stage 2
Regul.
Stage 3
Regul.
Stage 4
Regul.
Classification
Output

Figure 5. Neural network architecture for classification.

For each touch k in the training dataset made of K elements, a one-hot label $\mathbf{y}_k \in [0,1]^{N^2}$ is generated and corresponds to the zone that includes the touch position $\mathbf{t}_k \in [0,20] \times [0,20]$. The cross-entropy loss function is computed between the target $\mathbf{y}_k$ and the prediction $\hat{\mathbf{y}}_k = f(\mathbf{x}_k|\theta)$, where $f(\cdot|\theta)$ stands for a neural network with parameters

θ, and $\mathbf{x}_k$ is the measured signals in the time or frequency domains. The optimal parameters θ are obtained during training using the Adam optimizer. At test time, the neural network predicts the vector $\hat{\mathbf{y}}_m$ for each m out of M data points, from which the indices $(i,j)^* = \arg\max_{(i,j)} \hat{y}_{i,j}$ are obtained. The estimated touch position $\hat{\mathbf{t}}_m \in [0,20] \times [0,20]$ then corresponds to the center of mass of this zone, denoted as $\hat{\mathbf{t}}_m = \mathbf{c}_{(i,j)^*}$.

4.2. Regression

The regression approach aims at estimating directly the touch contact position. A four-stage fully connected neural network similar to the one introduced for classification is proposed in Figure 6. The architecture is identical, except for the output stage that contains only a linear layer that outputs a tensor $\hat{\mathbf{y}} \in \mathbb{R}^2$, which holds the horizontal and vertical positions of the touch. For each touch k in the training dataset, the target $\mathbf{y}_k \in \mathbb{R}^2$ corresponds to the touch position $\mathbf{t}_k \in [0,20] \times [0,20]$. The mean square error (MSE) loss is computed between the target $\mathbf{y}_k$ and the prediction $\hat{\mathbf{y}}_k = f(\mathbf{x}_k|\theta)$, where $f(\cdot|\theta)$ stands for the neural network with parameters θ, and $\mathbf{x}_k$ is the measured signal in the time or frequency domain. The estimated touch position $\hat{\mathbf{t}}_m \in [0,20] \times [0,20]$ then corresponds to the prediction $\hat{\mathbf{y}}_k$.

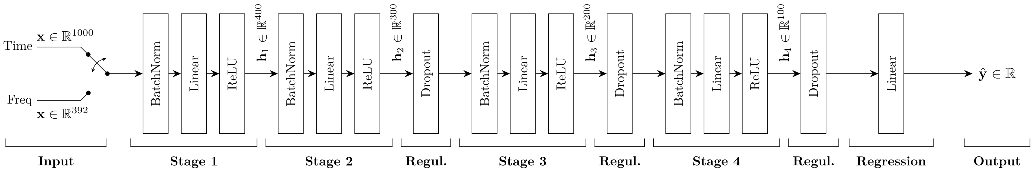
Time
$\mathbf{x} \in \mathbb{R}^{1000}$
Freq
$\mathbf{x} \in \mathbb{R}^{392}$
BatchNorm
Linear
ReLU
$\mathbf{h}_1 \in \mathbb{R}^{400}$
BatchNorm
Linear
ReLU
$\mathbf{h}_2 \in \mathbb{R}^{300}$
Dropout
BatchNorm
Linear
ReLU
$\mathbf{h}_3 \in \mathbb{R}^{200}$
Dropout
BatchNorm
Linear
ReLU
$\mathbf{h}_4 \in \mathbb{R}^{100}$
Dropout
Linear
$\hat{\mathbf{y}} \in \mathbb{R}$
Input
Stage 1
Stage 2
Regul.
Stage 3
Regul.
Stage 4
Regul.
Regression
Output

Figure 6. Neural network architecture for regression.

4.3. Results

The classification and regression approaches are validated with the test dataset. The localization performance is evaluated by comparing the estimated touch position and the baseline, using the Root Mean Square Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \|\hat{\mathbf{t}}_m - \mathbf{t}_m\|_2^2}, \qquad (1)$$

where $\|\cdot\|_2$ stands for the l^2 norm, and M corresponds to the number of data points in the test dataset. Note that the RMSE metrics penalize equally the position error in the horizontal and vertical dimensions. Figure 7 shows the RMSE (cm) as a function of the grid resolution N when using classification, and the RMSE when performing regression. The RMSEs are presented in time (blue) and frequency (red) domains. A 2 × 2 grid provides a mean localization error of 2.18 cm (standard deviation: 0.85 cm) and 2.1 cm (standard deviation: 0.8 cm) with frequency and time domains features, respectively. The RMSE reduces considerably when the classification touch zones are increased from 4 to 100 grids, as the center of mass of each class converges to the exact touch position when the grid resolution increases. With a 10 × 10 grid, the classification accuracy of 90% is achieved, which leads to a mean localization error of 0.41 cm (standard deviation: 0.25 cm) and 0.4 cm (standard deviation: 0.24 cm) with frequency and time domain features, respectively. The regression approach provides the mean localization error of 0.44 cm (standard deviation: 0.2 cm) and 0.45 cm (standard deviation: 0.21 cm) with frequency and time domains features, respectively. The results demonstrate that the RMSEs in the frequency and time domains are in the same range.

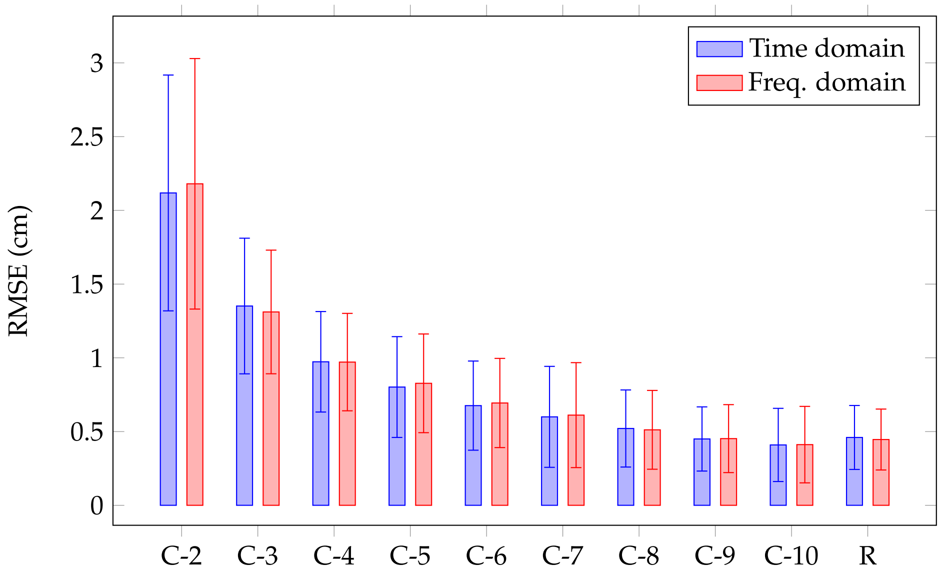

Figure 7. Comparison of localization RMSE (cm) between classification with grid resolution N (C-N), and regression (R) in frequency and time domains.

5. Applications

The results so far aimed at measuring and comparing the localization accuracy with respect to the model architecture and grid resolution. However, in a real scenario, the localization accuracy depends on the interface layout, and it is therefore important to measure the performance of the proposed architecture in such a realistic configuration. Thus far, the data were collected using a robot, which aims to mimic a human finger, but slightly differs due to material difference and pressure variation. This section shows how the proposed method performs with a virtual keypad interface, and illustrates the localization accuracy when a human user draws a shape with his finger. The feature extraction in frequency domain will be taken into consideration since it reduces the number of input parameters (392) compared to the time domain approach (1000).

5.1. Touch Localization on a Virtual Keypad Interface

The classification model previously introduced can be adapted and used to detect the touch coordinates on a virtual access control keypad, such as the one shown in Figure 8. This approach is appealing as the neural network is optimized to detect the keys for a given layout, which maximizes the localization accuracy. The same model architecture and dataset as the one formerly used for classification is chosen, except only the first two-stage fully connected neural network are considered. The first stage outputs the tensor $\mathbf{h}_1 \in \mathbb{R}^{100}$, while the second stage generates $\mathbf{h}_2 \in \mathbb{R}^{50}$. The output stage generates a tensor with 13 dimensions, where the first 12 classes correspond to the keys (1, 2, 3, 4, 5, 6, 7, 8, 9, *, 0 and #), and the last class represents the zone surrounding the keys (L).

The neural network predicts a vector with 13 elements, and the index of the element with the maximum value corresponds to the selected class. The confusion matrix in Figure 9 shows the performance of the classifier. According to the classification report, the overall accuracy is 97% with a computing time of 0.28 ms, using CPU (Intel Core i5). In this type of application, it is critical to avoid misclassification of a pressed key with another key. On the other hand, misclassifying a pressed key for the layout zone is less critical as the touch is simply ignored by the interface. The matrix diagonal represents the correct predictions as the true and predicted locations match. The proposed method classifies the pressed keys with an accuracy of 97%. There is no misclassification of a pressed key with another key. However, the neural network confuses some keys with the layout zone (class L), which can be disregarded by the interface.

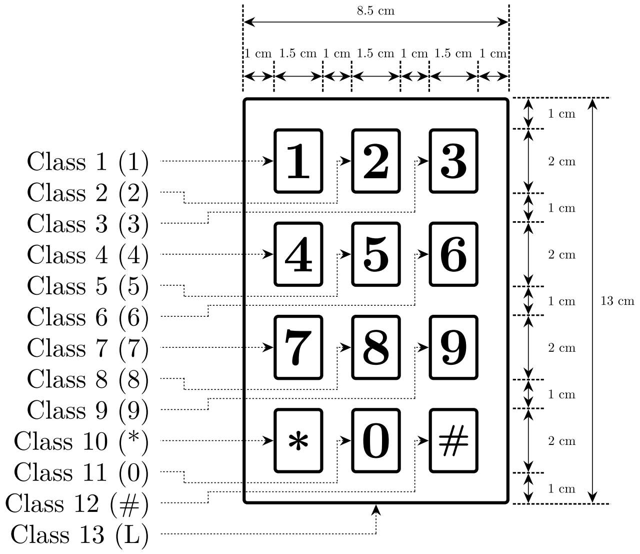

Figure 8. Access control keypad layout.

Target

Prediction	1	2	3	4	5	6	7	8	9	*	0	#	L
1	16	0	0	0	0	0	0	0	0	0	0	0	2
2	0	15	0	0	0	0	0	0	0	0	0	0	0
3	0	0	17	0	0	0	0	0	0	0	0	0	0
4	0	0	0	15	0	0	0	0	0	0	0	0	2
5	0	0	0	0	15	0	0	0	0	0	0	0	1
6	0	0	0	0	0	12	0	0	0	0	0	0	3
7	0	0	0	0	0	0	12	0	0	0	0	0	4
8	0	0	0	0	0	0	0	18	0	0	0	0	2
9	0	0	0	0	0	0	0	0	15	0	0	0	1
*	0	0	0	0	0	0	0	0	0	16	0	0	2
0	0	0	0	0	0	0	0	0	0	0	15	0	0
#	0	0	0	0	0	0	0	0	0	0	0	19	2
L	1	0	1	1	2	1	2	2	0	0	3	0	410

Figure 9. Confusion matrix with the access control keypad.

To provide a basis for comparison with potentially simpler approaches, the ML approach kNN (with k = 1) [33] has also been used to detect the touch coordinates for the access control keypad. Figure 10 shows the accuracy of the DNN and kNN approaches using 100%, 80%, 60%, 40%, and 20% of the training data set. For example, D-20 corresponds to 20% of the training data set. The accuracy drops from 97% (96%) to 89% (87%) when the size of data are decreased by 80% using DNN (kNN). Accuracy of the DNN method is always greater than the kNN considering different sizes of the dataset. This shows an improvement in test accuracy as the training set is enlarged.

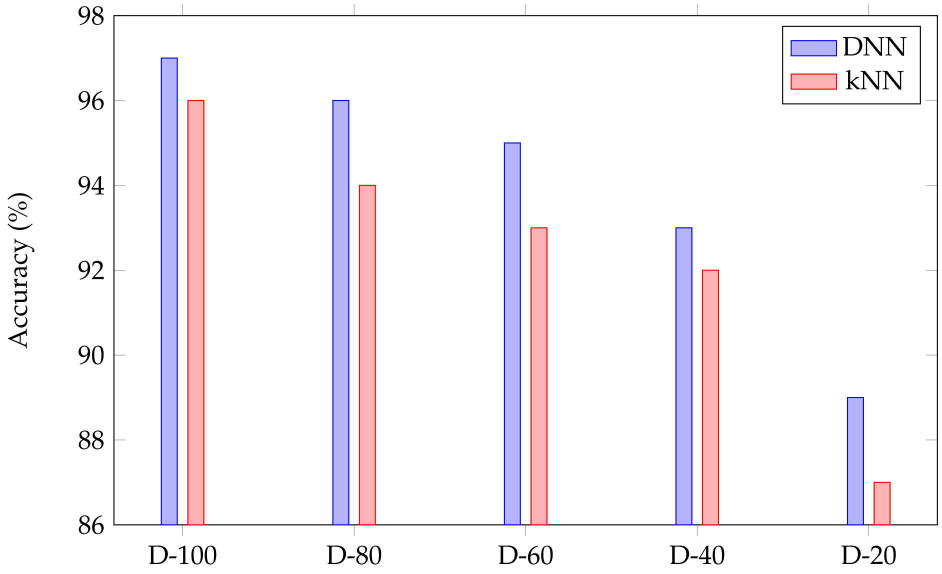

Figure 10. Comparison of accuracy between DNN and kNN approaches considering different data size, N% of the data (D-N) for the access control keypad.

Figure 11 compares the computing time between DNN and kNN approaches. These results indicate that the DNN approach is 11.67 times faster than the kNN approach. In the kNN approach, compressing the training set will reduce the time (by 2.58 ms) needed to search for the number of neighboring points, thus speeding up the process. On the other hand, this leads to decreased accuracy as shown in Figure 10. In DNN, expanding the data improves the accuracy without increasing the computational time. Thus, the DNN approach is more effective in terms of accuracy and computational time. These observations motivated the choice of the DNN approach over the kNN approach.

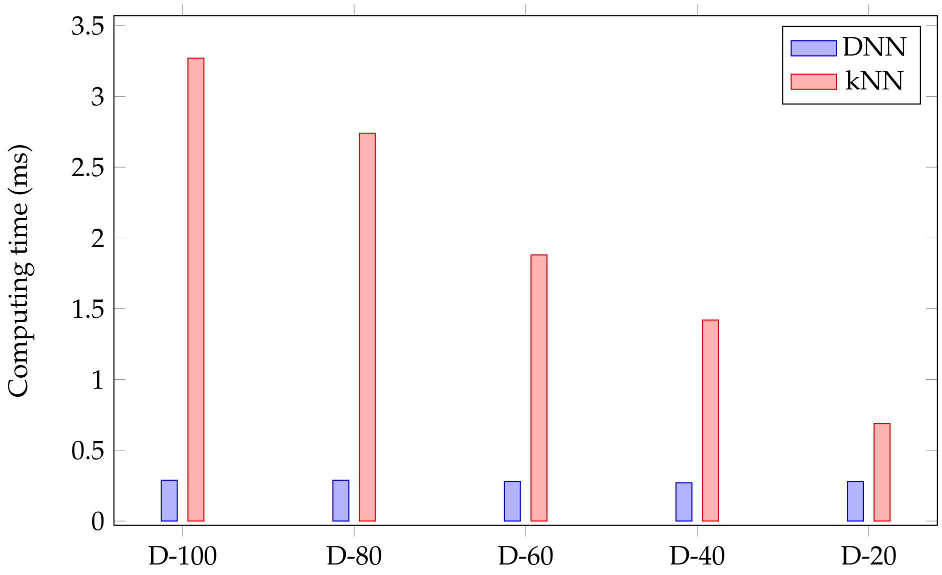

Figure 11. Comparison of computing time between DNN and kNN approaches considering different data size, N% of the data (D-N) for the access control keypad.

5.2. Localization of a Human Finger

The proposed architectures with classification (Figure 5) and regression (Figure 6) are also validated with a real human finger that draws a circle on the touch glass. Figures 12 and 13 show the human finger localization with classification with 10 grid resolution (C-10) and regression approaches, respectively.

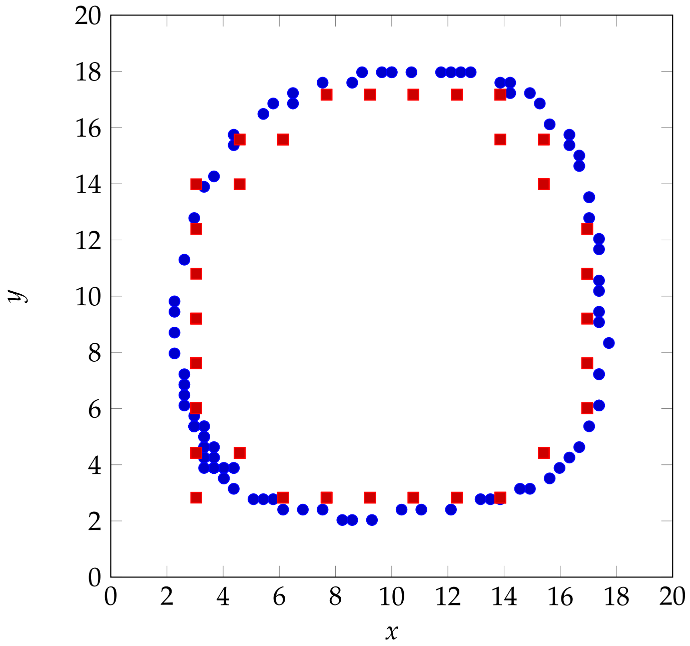

Figure 12. Human finger localization when using classification (C-10): baseline (blue) vs. prediction (red).

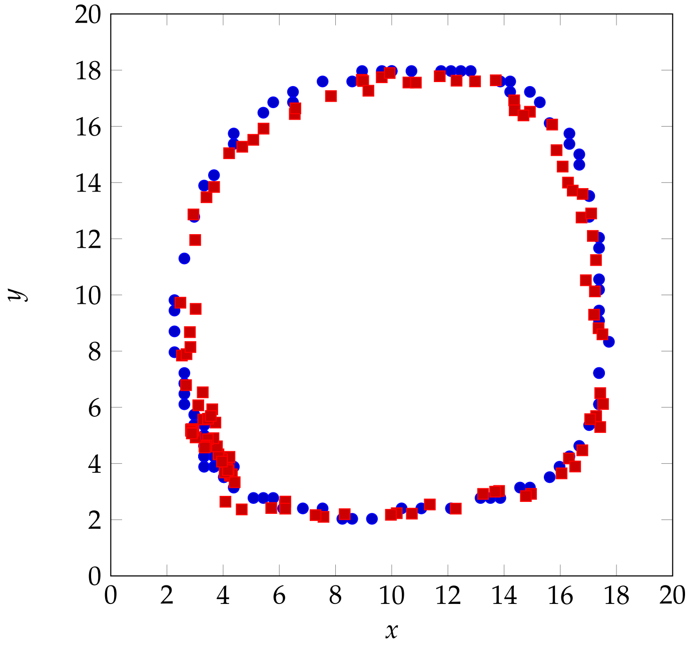

Figure 13. Human finger localization when using regression (R): baseline (blue) vs. prediction (red).

The blue dots show the true position of the finger and the red squares indicate the predicted positions. Figure 14 presents the localization RMSE using classification (C-5 and C-10) and regression.The regression approach provides the mean localization error of 0.47 cm, with the standard deviation of 0.18 cm, which leads to a minimum error of 0.29 cm, which is in the range of localization error in Figure 7. However, mean localization error is 0.69 cm, with the standard deviation of 0.29 cm using classification with 10 grid resolution (100 classes) and exceeds 1 cm (with the standard deviation of 0.54 cm) when the number of classes drops to 25. The computing time is 0.44 ms for each test sample. The small amount of training samples in each class enlarges the RMSE when using classification. Moreover, the localization error is more pronounced when the grid resolution reduces as the center of mass of each class diverges from the exact touch positions. This can explain the poor performance of C-5 and C-10 as shown in Figure 7. Regression overcomes these issues and also offers the best results.

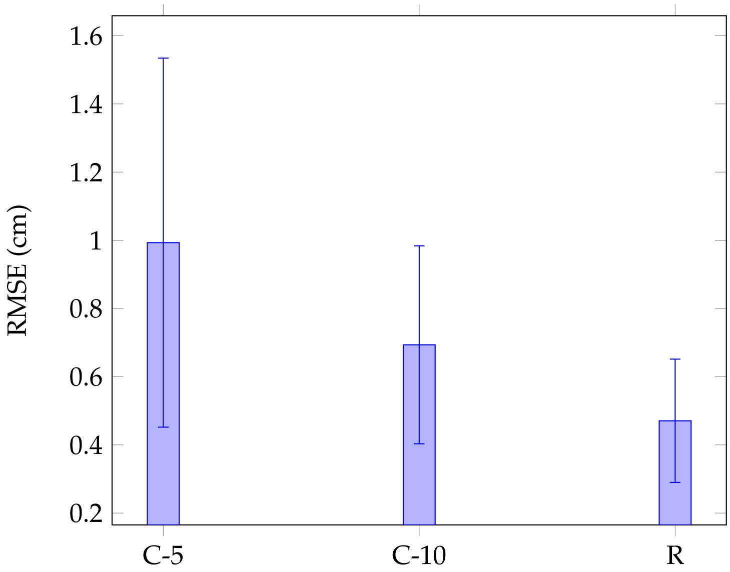

Figure 14. Comparison of human finger localization RMSE (cm) between classification with grid resolution N (C-N), and regression (R) in the frequency domain.

6. Conclusions

Localization techniques with classification and regression have been investigated on a glass touch surface with ultrasonic Lamb waves technology. The five piezoceramic elements are used as one emitter and four receivers installed at the bottom of the surface. In this study, a simple fully connected neural network is proposed to perform touch localization on the glass plate. The aim is to reduce the computational complexity of the analytical imaging approaches associated with touch localization techniques. A robotic arm with a silicon finger simulates the touch action with random position and contact pressure to train the deep neural network. Frequency-domain features are selected as it reduces significantly the number of input parameters compared to a time domain approach. The proposed processing architecture is then validated with a human finger to localize the touches, which leads to a mean error of 0.47 cm and standard deviation of 0.18 cm. The computing time is 0.44 ms for each test sample. The classification approach is applied for touch detection on an access control keypad, which provides an accuracy of 97% with a computing time of 0.28 ms for each unseen example.

During the data acquisition, the human fingers can be swiped on the screen and the touch pressure can change. This sets a limit on the accuracy of a signal measurement when gathering human finger data. The difference between signals generated by the human finger and the artificial finger should be taken into account. The human finger and artificial finger localization errors (presented in Figures 7 and 14, respectively) are in the same range. The results validate the similarity between the signals created by the simulated touch and those generated by the real touch. The performance of the classification for the touch localization is, however, limited by grid resolution. As the grid resolution decreases, the center of mass of each class moves away from exact touch position, decreasing the localization accuracy. As shown in Figure 7, increasing the grid resolution improves the localization accuracy for the classification approach. Under such circumstances, the regression remains valid to localize the touches.

This analysis demonstrates the viability of the regression with a four-stage fully connected neural network for touch localization on an ultrasonic Lamb wave touchscreen. The classification with a two-stage fully connected neural network is preferred for the touch zone detection due to its ability to provide high-precision touch detection on an access control keypad. The results indicate that a current analytical, and computationally intensive, touch localization algorithm with a simple fully connected layer is possible.

In future work, this could be extended to multi-touch scenarios. Multi-touch artificial fingers can be designed to acquire multi-touch signals. The double-touch simulator is shown in [5]. The other possibility would be training and testing the model with human fingers. This could provide more accurate results, but, in return, may require more time to collect the data. Moreover, the robustness of the model can be affected by the touch pressure of human fingers. In multi-touch gestures, it is critical that all the fingers will be pressed and released while taking data to avoid mislabeling of the number of touches. The proposed model in the current study is easily scalable to surfaces with ultrasonic Lamb waves technology of different shape, size and material, including not being limited to plastic and metal.

Author Contributions: Conceptualization, S.B., J.M., P.M. and F.G.; Data curation, S.B. and J.M.; Formal analysis, S.B., J.M. and F.G.; Funding acquisition, J.M., P.M. and F.G.; Investigation, S.B.; Methodology, S.B. and F.G.; Project administration, J.M., P.M. and F.G.; Resources, J.M. and P.M.; Software, S.B. and J.M.; Supervision, P.M. and F.G.; Validation, S.B. and F.G.; Visualization, S.B.; Writing—original draft, S.B., J.M., P.M. and F.G.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Mitacs (Canada).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Walker, G. A review of technologies for sensing contact location on the surface of a display. *J. Soc. Inf. Disp.* **2012**, *20*, 413–440. [CrossRef]
2. Dai, J.; Chung, C. Touchscreen everywhere: On transferring a normal planar surface to a touch-sensitive display. *IEEE Trans. Cybern.* **2013**, *44*, 1383–1396. [CrossRef]
3. Bhalla, M.; Bhalla, A. Comparative study of various touchscreen technologies. *Int. J. Comput. Appl.* **2010**, *6*, 12–18. [CrossRef]
4. Quaegebeur, N.; Masson, P.; Beaudet, N.; Sarret, P. Touchscreen surface based on interaction of ultrasonic guided waves with a contact impedance. *IEEE Sens. J.* **2016**, *16*, 3564–3571. [CrossRef]
5. Yang, Z.; Liu, X.; Wu, B.; Liu, R. Adaptability of Ultrasonic Lamb Wave Touchscreen to the Variations in Touch Force and Touch Area. *Sensors* **2021**, *21*, 1736 [CrossRef]
6. Adler, R.; Desmares, P. An economical touch panel using SAW absorption. *IEEE Trans. Ultrason. Ferroelectr. Addit. Freq. Control.* **1987**, *34*, 195–201. [CrossRef]
7. Son, K.; Lee, C. Design and reliability of acoustic wedge transducer assemblies for outdoor touch panels. *IEEE Trans. Compon. Packag. Addit. Manuf. Technol.* **2011**, *1*, 1178–1185. [CrossRef]
8. Ing, R.; Quieffin, N.; Catheline, S.; Fink, M. In solid localization of finger impacts using acoustic time-reversal process. *Appl. Phys. Lett.* **2005**, *87*, 204104. [CrossRef]
9. Liu, Y.; Nikolovski, J.; Mechbal, N.; Hafez, M.; Vergé, M. Tactile objects based on an amplitude disturbed diffraction pattern method. *Appl. Phys. Lett.* **2009**, *95*, 251904. [CrossRef]
10. Liu, Y.; Nikolovski, J.; Mechbal, N.; Hafez, M.; Vergé, M. An acoustic multi-touch sensing method using amplitude disturbed ultrasonic wave diffraction patterns. *Sens. Addit. Actuators A Phys.* **2010**, *162*, 394–399. [CrossRef]
11. Pham, D.; Ji, Z.; Yang, M.; Wang, Z.; Al-Kutubi, M. A novel human-computer interface based on passive acoustic localisation. In Proceedings of the International Conference On Human-Computer Interaction, Beijing, China, 22–27 July 2007; pp. 901–909.
12. Firouzi, K.; Nikoozadeh, A.; Khuri-Yakub, B. Numerical modeling of ultrasonic touchscreen. In Proceedings of the IEEE International Ultrasonics Symposium, Chicago, IL USA, 3–6 September 2014; pp. 753–756.
13. Firouzi, K.; Nikoozadeh, A.; Carver, T.; Khuri-Yakub, B. Lamb wave multitouch ultrasonic touchscreen. *IEEE Trans. Ultrason. Ferroelectr. Addit. Freq. Control* **2016**, *63*, 2174–2186. [CrossRef] [PubMed]
14. Campbell, C. *Surface Acoustic Wave Devices and Their Signal Processing Applications*; Elsevier: Amsterdam, The Netherlands, 2012.
15. Reis, S.; Correia, V.; Martins, M.; Barbosa, G.; Sousa, R.; Minas, G.; Lanceros-Mendez, S.; Rocha, J. Touchscreen based on acoustic pulse recognition with piezoelectric polymer sensors. In Proceedings of the IEEE International Symposium On Industrial Electronics, Bari, Italy, 4–7 July 2010; pp. 516–520.
16. Reju, V.; Khong, A.; Sulaiman, A. Localization of taps on solid surfaces for human-computer touch interfaces. *IEEE Trans. Multimed.* **2013**, *15*, 1365–1376. [CrossRef]

17. North, K.; D Souza, H. Acoustic pulse recognition enters touch-screen market. *Inf. Disp.* **2006**, *22*, 22.
18. Ing, R.; Cassereau, D.; Fink, M.; Nikolovski, J. Tactile touch plate with variable boundary conditions. *J. Acoust. Soc. Am.* **2008**, *123*, 4225–4229. [CrossRef]
19. Masson, P.; Quaegebeur, N.; Ostiguy, P.; Beaudet, N.; Sarret, P. Active Acoustic Pressure Mapping System. U.S. Patent 9,750,451, 5 September 2017.
20. Firouzi, K.; Khuri-Yakub, B. A learning method for localizing objects in reverberant domains with limited measurements. *J. Acoust. Soc. Am.* **2017**, *141*, 104–115. [CrossRef]
21. Chang, C.; Lee, Y. Ultrasonic Touch Sensing System Based on Lamb Waves and Convolutional Neural Network. *Sensors* **2020**, *20*, 2619. [CrossRef]
22. Li, C. *Using Ray Tracking and Machine Learning to Localize a Lamb Wave Scatterer on a Plate*; Hong Kong University of Science: Hong Kong, China, 2020.
23. Alleyne, D.; Cawley, P. Optimization of Lamb wave inspection techniques. *NDT E Int.* **1992**, *25*, 11–22. [CrossRef]
24. Feng, B.; Pasadas, D.; Ribeiro, A.; Ramos, H. Locating defects in anisotropic CFRP plates using ToF-based probability matrix and neural networks. *IEEE Trans. Instrum. Addit. Meas.* **2019**, *68*, 1252–1260. [CrossRef]
25. Xu, B.; Yu, L.; Giurgiutiu, V. Advanced methods for time-of-flight estimation with application to Lamb wave structural health monitoring. In Proceedings of the International Workshop On Structural Health Monitoring, Stanford, CA, USA, 9–11 September 2009; pp. 1202–1209.
26. Cantero-Chinchilla, S.; Chiachío, J.; Chiachío, M.; Chronopoulos, D.; Jones, A. A robust Bayesian methodology for damage localization in plate-like structures using ultrasonic guided-waves. *Mech. Syst. Addit. Signal Process.* **2019**, *122*, 192–205. [CrossRef]
27. Giurgiutiu, V.; Zagrai, A.; Jing Bao, J. Piezoelectric wafer embedded active sensors for aging aircraft structural health monitoring. *Struct. Health Monit.* **2002**, *1*, 41–61. [CrossRef]
28. Torkamani, S.; Roy, S.; Barkey, M.; Sazonov, E.; Burkett, S.; Kotru, S. A novel damage index for damage identification using guided waves with application in laminated composites. *Smart Mater. Addit. Struct.* **2014**, *23*, 095015. [CrossRef]
29. Quaegebeur, N.; Masson, P.; Langlois-Demers, D.; Micheau, P. Dispersion-based imaging for structural health monitoring using sparse and compact arrays. *Smart Mater. Addit. Struct.* **2011**, *20*, 025005. [CrossRef]
30. Quaegebeur, N.; Ostiguy, P.; Masson, P. Correlation-based imaging technique for fatigue monitoring of riveted lap-joint structure. *Smart Mater. Addit. Struct.* **2014**, *23*, 055007. [CrossRef]
31. Sbarufatti, C.; Manson, G.; Worden, K. A numerically-enhanced machine learning approach to damage diagnosis using a Lamb wave sensing network. *J. Sound Addit. Vib.* **2014**, *333*, 4499–4525. [CrossRef]
32. De Fenza, A.; Sorrentino, A.; Vitiello, P. Application of Artificial Neural Networks and Probability Ellipse methods for damage detection using Lamb waves. *Compos. Struct.* **2015**, *133*, 390–403. [CrossRef]
33. Vitola, J.; Pozo, F.; Tibaduiza, D.; Anaya, M. A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications. *Sensors* **2017**, *17*, 417. [CrossRef]
34. Zhang, Z.; Pan, H.; Wang, X.; Lin, Z. Machine learning-enriched lamb wave approaches for automated damage detection. *Sensors* **2020**, *20*, 1790. [CrossRef]
35. Yang, L.; Xu, F. A novel acoustic emission sources localization and identification method in metallic plates based on stacked denoising autoencoders. *IEEE Access* **2020**, *8*, 141123–141142. [CrossRef]
36. Grondin, F.; Glass, J. Multiple sound source localization with SVD-PHAT. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 2698–2702.
37. Grondin, F.; Glass, J. SVD-PHAT: A fast sound source localization method. In Proceedings of the IEEE International Conference On Acoustics, Speech In addition, Signal Processing, Brighton, UK, 12–17 May 2019; pp. 4140–4144.
38. Rautela, M.; Senthilnath, J.; Moll, J.; Gopalakrishnan, S. Combined two-level damage identification strategy using ultrasonic guided waves and physical knowledge assisted machine learning. *Ultrasonics* **2021**, *115*, 106451. [CrossRef]
39. Ewald, V.; Groves, R.; Benedictus, R. DeepSHM: A deep learning approach for structural health monitoring based on guided Lamb wave technique. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019, Denver, CO, USA, 4–7 March 2019; Volume 10970, p. 109700H.
40. De Oliveira, M.; Monteiro, A.; Vieira Filho, J. A new structural health monitoring strategy based on PZT sensors and convolutional neural network. *Sensors* **2018**, *18*, 2955. [CrossRef]
41. Kim, I.; Jeon, H.; Baek, S.; Hong, W.; Jung, H. Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle. *Sensors* **2018**, *18*, 1881. [CrossRef] [PubMed]
42. Ye, J.; Ito, S.; Toyama, N. Computerized ultrasonic imaging inspection: From shallow for deep learning. *Sensors* **2018**, *18*, 3820. [CrossRef] [PubMed]
43. Civera, M.; Surace, C. Non-Destructive Techniques for the Condition and Structural Health Monitoring of Wind Turbines: A Literature Review of the Last 20 Years. *Sensors* **2022**, *22*, 1627. [CrossRef]
44. Zonzini, F.; Bogomolov, D.; Dhamija, T.; Testoni, N.; De Marchi, L.; Marzani, A. Deep Learning Approaches for Robust Time of Arrival Estimation in Acoustic Emission Monitoring. *Sensors* **2022**, *22*, 1091. [CrossRef] [PubMed]
45. Le, H.; Mayer, S.; Henze, N. Investigating the feasibility of finger identification on capacitive touchscreens using deep learning. In Proceedings of the International Conference On Intelligent User Interfaces, Marina del Ray, CA, USA, 17–20 March 2019; pp. 637–649.