Ultrasonics 119 (2022) 106598

Contents lists available at ScienceDirect

Ultrasonics

journal homepage: www.elsevier.com/locate/ultras

Detection of overlapping ultrasonic echoes with deep neural networks

Alon Shpigler [a,*], Etai Mor [b], Aharon Bar-Hillel [a]

[a] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
[b] Department of Non Destructive Testing, Soreq Nuclear Research Center, Yavne 81800, Israel

ARTICLE INFO

Keywords:
Overlapping echoes separation
Ultrasound
Layer thickness estimation
Detection
Inverse problems
Deep learning
Non destructive testing

ABSTRACT

Ultrasonic Pulse-Echo techniques have a significant role in monitoring the integrity of layered structures and adhesive joints along their service life. However, when acoustically measuring thin layers, the resulting echoes from two successive interfaces overlap in time, limiting the resolution that can be resolved using conventional pulse-echo techniques. Deep convolutional networks have arisen as a promising framework, providing state-of-the-art performance for various signal processing tasks. In this paper, we explore the applicability of deep networks for detection of overlapping ultrasonic echoes. The network is shown to outperform traditional algorithms in simulations for a significant range of echo overlaps, echo pattern variance and noise levels. In addition, experiments on two physical phantoms are conducted, demonstrating superiority of the network over traditional methods for layer thickness estimation.

1. Introduction

Ultrasonic pulse-echo methods have been widely used in the analysis of thickness and bond quality of layered materials in the Non-Destructive Testing (NDT) domain [1]. With these methods, an ultrasonic transducer sends an ultrasonic pulse and receives the reflected echoes from interfaces in the inspected sample. The arrival time of reflected echoes indicate the position of the interfaces, while amplitudes are used to evaluate the interface condition. However, the performance of these methods is limited for thick layers in which successive echoes do not overlap in time. Such overlap occurs when the time-of-flight difference (TOFD) in the layer is shorter than the pulse width, forming an axial resolution problem. In this case, simple derivation of individual echo parameters is not possible. An additional challenge often posed in this respect is reverberating waves from the inspected layers, forming successive overlaps.

Large overlaps between echoes create a difficult pattern recognition problem. First, it is difficult to understand how many echoes are involved. Second, it is difficult to determine their exact time-of-flight (TOF). An illustration of several difficult cases of overlap detection is shown in Fig. 1. One way to overcome this problem would be the use of high frequency, wideband transducers, which reduce the temporal pulse width and therefore enhance the resolution between successive echoes. However, due to increased attenuation, higher frequencies provide less penetration through the material layers.

Another difficulty arises when echoes are not identical, and their shape is not fully known a-priory. While the echoes are similar to a prototypical pulse originally sent by the transducer, they may have wide variance in their phase, center frequency, width and amplitude. These variations are caused by several physical reasons, including frequency depended attenuation and dispersion.

Numerous methods were suggested over the past several decades to solve this task. A simple and classical technique for solving overlapping echoes is by the amplitude spectrum method [2,3], where a signal containing overlapping echoes is separated by the minima in the frequency-domain. This method is suitable for estimating the TOFD of a single layer embedded between two thick layers. However, it is not situated for separation of echoes caused by several unknown layers, and can also be sensitive to noise, especially when the resonance minima is at the high end of the bandwidth where the Signal-to-Noise Ratio (SNR) is low. Another approach is model-based signal decomposition by estimation of the echo parameters [4,5]. While this approach is rather simple and does not rely on a reference signal, it can only converge to a local minimum, thus is highly depended on a good initialization. Moreover, this approach usually requires the number of echoes, composing the signal, to be known in advance.

Another popular approach is using sparse signal representation (SSR) methods, otherwise known as dictionary-based methods [6–9]. These techniques have been widely used for ultrasound echoes separation thanks to their efficiency in adaptive signal decomposition [1]. A major limitation of these techniques arises with growing variance in the echo patterns. To enable successful detection, the dictionary might require multiple values for each parameter, and a Cartesian

* Corresponding author.
E-mail address: alonshp@post.bgu.ac.il (A. Shpigler).

https://doi.org/10.1016/j.ultras.2021.106598
Received 11 May 2021; Received in revised form 14 September 2021; Accepted 21 September 2021
Available online 9 October 2021

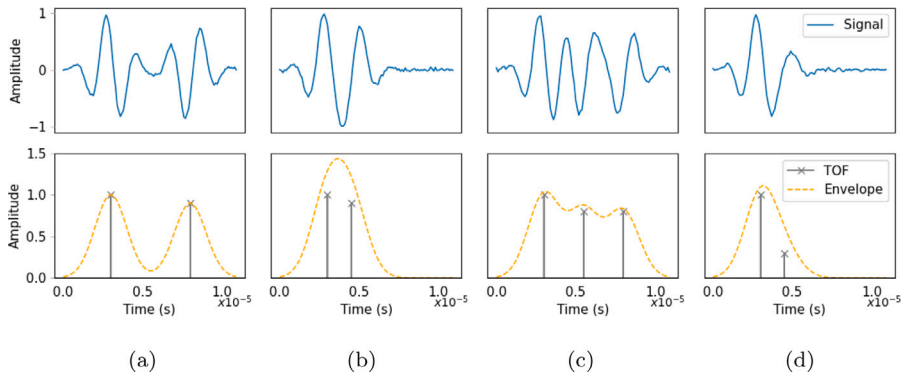

Fig. 1. Illustrations of ultrasonic overlapping echoes (top) along with their envelopes and TOFs (bottom). (a) Minor overlap, (b) major overlap, (c) successive overlaps, and (d) overlap with a low-amplitude echo.

product of the values used for multiple parameters. Dictionary size hence grows exponentially in the number of parameters, which leads to slow detection and higher false alarm rate.

Convolutional neural networks (CNNs) [10] have emerged as a promising direction for a variety of complex pattern recognition problems, and in recent years were adopted in the signal processing domain to solve sequence processing problems like text-to-speech [11], music synthesis [12,13] and speech enhancement [14–16]. Contrary to the classical inverse methods which rely solely on a proximal physical model and simple priors, these methods are learned from collected data in which ground-truth reconstructed signals are known for measurement examples. They hence enable learning more complex data-dependent priors and predictions. In image recognition, convolution layers were shown to successfully model complex features and object parts [17]. Similarly, such layers can potentially model echo parts and overlapping echoes, and learn to reconstruct overlaps with different shapes.

Fundamentally, the basic building blocks in CNN architecture, convolutions, are well suited for the inverse operation taking a signal to its generating delta function — a deconvolution task. In previous studies, networks were shown to successfully embed traditional inverse iterative algorithms into layered CNNs [18,19]. In addition, CNNs can be effective for modeling noise inflicted by detector sensitivity and exhaustion, as demonstrated by signal denoising networks [14–16]. Recently Li et al. [20] utilized a deep learning framework to separate overlapping ultrasound echoes and estimate the thickness of buried pipelines. Their solution is shown to improve TOFD estimation, however it suffers from several limitations. In their work, two networks are trained independently, solving several separate sub-problems. The first network segments the signal, including two overlapping echoes, into two separate signals. The second network is applied to each signal in isolation and finds the echo's TOF. Both networks used are rather heavy-weighted considering the needs of the problem, rendering slow training and inference.

In this paper, we suggest a simple network which solves the problem end-to-end as a detection problem. We construct a lean, yet effective, deep network for the overlapping echoes detection task. Unlike the method of [20], which is limited to two overlapping echoes, our proposed network solves a general detection problem, with unknown number of overlapping echoes and varying echo patterns. The proposed CNN framework is composed of long filters in the first layer to simulate the deconvolution of the signal back to its origin delta, followed by several layers with short filters to project the deconvolved activations to the space of clean and sparse sequences with few active delta functions. The training process is conducted with a simulated dataset, created based on a physical model of the ultrasound echoes, with significant variance in echo parameters, noise and overlaps.

The proposed network is compared to competing methods on simulated test datasets and on two physical phantoms taken from different materials. The network achieves high detection rates in adverse conditions including high echo pattern variance and severe successive overlaps. Our simulation experiments show that the network is preferable to competing algorithms in a wide range of signal overlap and SNR levels, and enables more accurate detection. With the physical phantoms, the network was tested and compared to competing algorithms at the task of layer thickness estimation from ultrasonic scans. The results show the network enables higher axial resolution, and provides more stable and accurate prediction then competing methods.

2. Problem formulation and related work

Traditional signal processing methods model the measured signal $y(t)$ using a linear time-invariant convolutional model of the form

$$y(t) = d(t) * x(t) + e(t), \tag{1}$$

where $*$ denotes the linear convolution operator, $d(t)$ is the transmitted ultrasonic pulse, $x(t)$ is the inspected object reflectivity, and $e(t)$ is a Gaussian noise function. Deconvolution of $x(t)$ given $y(t)$ and $d(t)$ is an ill-conditioned problem since $d(t)$ is usually a narrow-band signal. Therefore, one should enforce some prior on $x(t)$ to obtain de-convolution results with a physical meaning. For the ultrasound application, strong echoes are expected only from interface locations between two materials with different impedance. Hence $x(t)$ is comprised mainly of zeros, and a sparsity constraint on $x(t)$ makes a good prior. The common way to enforce this constraint is by using the "sparse spike train" model [8,21,22]

$$y(t) = d(t) * \left\{ \sum_{m=1}^{M} x_m \delta(t - \tau_m) \right\} + e(t), \tag{2}$$

where $\delta(t)$ is the Dirac delta-function, M is the number of echoes, and x_m, τ_m are the amplitude and time of flight of the mth echo. Model (2) defines all the echoes to be of the same form $x_m d(t - \pi_m)$, hence it can only be used when the attenuation is negligible and pulse shape does not significantly vary with time. A more flexible model [4], where each individual echo may have a unique shape, can be defined by

$$y(t) = \sum_{m=1}^{M} x_m d(\boldsymbol{\theta}_m, t - \tau_m) + e(t). \tag{3}$$

Here, $d(\boldsymbol{\theta}_m, t)$ represents the mth ultrasonic pulse shape, modeled by the parameters $\boldsymbol{\theta}_m$, and pulse shape is allowed to vary among echos.

Traditionally, $\{x_m, \tau_m, \boldsymbol{\theta}_m\}_{m=1}^{M}$ is recovered from $y(t)$ using dictionary-based methods. This is achieved by seeking a sparse vector $\mathbf{x}$ satisfying the relationship $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e}$ where $\mathbf{D} \in C^{n \times m}$ is an overcomplete dictionary composed of $\{\boldsymbol{\Phi}_i\}_{i=1}^{m}$ atoms in columns, with $\boldsymbol{\Phi}_i \in C^n$ and $m > n$. Matching Pursuit (MP) [6] greedily constructs successive approximations of the signal by finding maximal projections of the signal residual on atoms of $\boldsymbol{D}$. Orthogonal Matching Pursuit (OMP) [7]

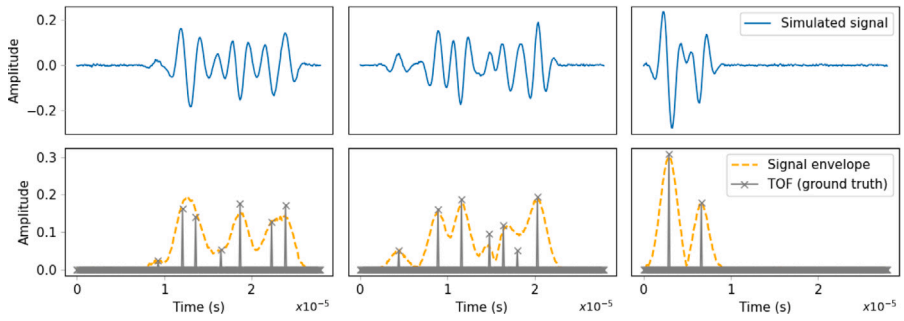

Fig. 2. Simulated data illustration. Examples of Simulated signals (top) used for network training and testing, along with their envelopes and TOFs (bottom).

improves MP by adding a least-square minimization to each step of MP. Support Matching Pursuit (SMP) [8], specifically designed for resolving ultrasonics overlapping echoes, extends MP by adding in the iterative stage a relaxed support measure corresponding to the p-norm with $0 < p < 1$. Another method for solving sparse problems is the Iterative Shrinkage Thresholding Algorithm (ISTA) [9]. This algorithm iterates between two operations: linear estimation, and shrinkage based on a soft thresholding function.

3. Echo detection using deep networks

A network is trained for echo detection using a dataset $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^{N}$ of signal examples $\boldsymbol{y}_i \in R^{T_s}$ along with their respective ground truth delta-train reconstructions $\boldsymbol{x}_i \in R^{T_s}$. The network is a parametric inverse model $h_{\boldsymbol{W}}(\boldsymbol{y}) : R^{T_s} \rightarrow R^{T_s}$, accepting a signal example $\boldsymbol{y}$ as input and parametrized by a set of weight parameters $\boldsymbol{W}$. Training consists in minimizing a loss function over the choice of $\boldsymbol{W}$

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmin}} \sum_{i=1}^{N} L(h_{\boldsymbol{W}}(\boldsymbol{y}_i), \boldsymbol{x}_i) + J(\boldsymbol{W}). \tag{4}$$

Here L is a loss function between the ground truth and the model prediction, and J is a regularization term posing limitations on the model parameters to reduce overfit. In the following sections we describe the details of the dataset, the model $h_{\boldsymbol{W}}(y)$, and the training process.

3.1. Data simulation

The physical model used for generation of a single echo is described in 3.1.1, followed by description of signals and outputs in 3.1.2.

3.1.1. Gaussian echo model

Various parametric models have been proposed to model the echo pattern $d(\boldsymbol{\theta}_m, t)$ described in Eq. (3) [1,4,23,24]. We use a model describing the ultrasonic echo with parameters $\boldsymbol{\theta}_m = [\omega, \phi, \sigma]$, associated with physical parameters of the pulse as follows

$$d(\boldsymbol{\theta}, t-\tau) = K_\theta g(t-\tau, \sigma) cos[\omega(t-\tau) + \phi], \tag{5}$$

which is the product of a Gaussian envelop function with a harmonic function. $g(t-\tau, \sigma)$ is a Gaussian envelop, with σ defining the scale (proportional to the pulse-width) of the echo around its arrival time τ. $cos[\omega(t-\tau)+\phi]$ defines an harmonic function with ϕ and ω its phase and center-frequency. The constant K_θ ensures that $\|d(\boldsymbol{\theta}, t-\tau)\|_2 = 1$.

3.1.2. Dataset randomization

Signal examples $\boldsymbol{y}$ are created as superpositions (Eq. (3)) of Gaussian echoes (Eq. (5)). For each generated signal $\boldsymbol{y}$, a target sparse train of delta functions $x(t) = \sum_{m=1}^{M} x_m \delta(t-\tau_m)$ is generated as the reconstruction target. Examples of the simulated signals used for network training and testing are demonstrated in Fig. 2.

Echo Pattern Randomization. Echo pattern parameters described in Eq. (5), $\boldsymbol{\theta} = [\omega, \phi, \sigma]$, are drawn from uniform distributions. The phase ϕ is drawn in $[0, 2\pi]$. The center frequency ω and the signal width σ are drawn in $[\omega_0, \omega_r]$, $[\sigma_0, \sigma_r]$ respectively, where ω_0, σ_0 are the center frequency and pulse width measured from a transducer reference echo-signal received from a flat reflector immersed in water. The parameters ω_r, σ_r control the width of the echo distributions. Specific values and distributions are described in Section 4.

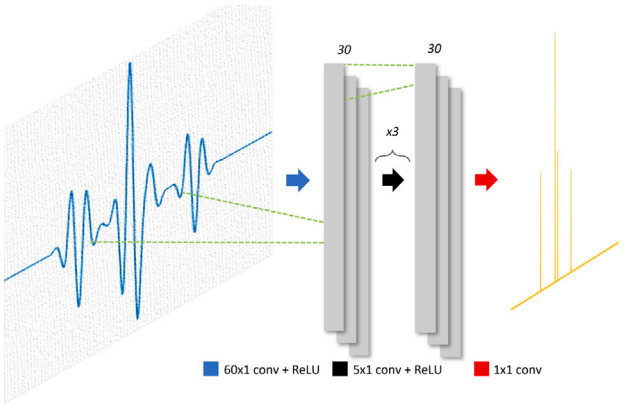

Fig. 3. Schematic overview of the suggested US-CNN architecture.

Full signal Randomization. The number of echoes in a signal is drawn from a uniform distribution in the range $[1, K]$. In order to create significant overlap between echoes in the dataset, the TOFD between neighboring echoes is drawn from an exponential distribution. The first echo TOF is randomly drawn uniformly from the T_s time-samples, and the following echoes' TOFD is drawn from $exp(\lambda)$. λ hence defines the task overlap difficulty, and can be adapted to the task needs. In our experiments, we set $\lambda = \frac{1}{3\sigma_0}$, and allow TOFD range of $[1.5\sigma_0, 6\sigma_0]$, with $\sigma_0 = 9$ time-samples. The echo amplitude x is drawn uniformly in $[\alpha, 1]$, with α a difficulty parameter controlling the echo's dynamic range. Noise is modeled in the signal by additive Gaussian noise. The level of noise is defined by $SNR = 10log_{10}(\frac{P_s}{P_n})$, with P_s the power of the signal (normalized to 1 in all experiments) and P_n is the power of the noise. The desired noise level is set by tuning P_n.

3.2. Network design

The network $h_W(y) : R^{T_s} \rightarrow R^{T_s}$ is a Fully Convolutional Neural Network (FCN) [25]. It receives as input the signal $y(t)$ of T_s time stamps, and outputs a prediction $x(t)$ of the same length. The proposed network, illustrated in Fig. 3, consists of two main parts: an inverse operator and a projection operator. The inverse operator is a single wide convolution layer, preforming correlations with multiple learned pulses of the signal with multiple pulse shapes. The projection is performed by several convolutional layers with small-sized filters, projecting the noisy result to the space of clean, sparse and separated deltas.

Inverse operator. Since the signal generation presented in Eq. (3) defines a superposition of echos with variance in their shape, its inverse function can be modeled by a standard convolution layer with multiple filters whose size is large enough to cover the echo's length. The wide convolution layer is employed with a filter size of $6\sigma_0$, the approximated length of the generating pulses, and depth f, the number of representative filters, set in the following experiments to 30. The input is padded with $3\sigma_0$ for the output to obtain the length of the input signal T_s. The wide convolution is followed by a ReLU [26] activation.

To observe the effect of first layer filters, the network was trained separately with different number of filters and representative filters were inspected. In Fig. 4, representative first layer filters are presented. In Fig. 4 (left), a filter from a network trained with a single filter is shown. With one filter, the network functions as an adaptive non-linear filter (with ReLU layers providing non-linearity). The filter learns a sinusoidal shape with a minor peak at its center, which is suitable to locate an echo. However, this alone is not sufficient to resolve overlapping echoes of different patterns, and indeed a network with a single filter produces poor overlap separation results (over a 55% decrease in echo detection accuracy compared to network results described in Section 5.1). In Fig. 4 (right), filters with the largest norms from a network with 30 filters in the first layer filters are presented. One can observe that the learned filters reflect the temporal oscillatory nature of the received signals related to the signal bandwidth and central frequency. Multiple filters adaptively and simultaneously learn to identify different time-variant oscillatory shapes, information that is later integrated by the network for the overlap separation prediction.

Projection operator. After deconvolution, multiple maps may exhibit strong response to the same echo reflection, as a single echo is detected by multiple filters. However, in the desired output only a single exact TOF of a reflection is represented by a non zero value. To enable reasoning, competition among neighboring components and removal of redundant noise, the inverse operator is followed by a series of *3* conv layers. These layers use a kernel of size *5*, stride *1* and dilation [11,16] of *2*, and each is followed by a ReLU activation. The last convolution layer is followed by a dropout layer [27], employed to improve network robustness to overfitting, with the dropout rate set to 0.2. Finally, the activations are transferred into 1×1 convolution to reduce the multiple maps representing each time-sample into a single output.

3.3. Network training

We pose echo detection as a dense regression task — for each time sample, the model predicts the amplitude of the echo reflection at that point, if there was any. Given N training signals $\{y_1, \ldots, y_N\}$, the network is trained using a dense l_2 loss

$$\mathcal{L}_{MSE} = \sum_{i=1}^{N} \sum_{t=1}^{T_s} (h_W(y_i(t)) - \hat{x}_i(t))^2. \tag{6}$$

where $\hat{x}_i$ are soft versions of x_i described below.

Gaussians for prediction relaxation. The ground truth $x(t)$ is a sparse vector, with few non zero values representing the echoes' TOFs. Estimation of such non-continuous target vectors is a difficult task to learn. Deviation of only one time-sample in prediction cause an error much bigger than if the prediction was zero, even though the prediction is almost precise. This discourages learning of useful, non trivially-zero predictions. To enable a smoother energy landscape for the learning problem, we propose to smooth $x(t)$ by convolving it with a zero mean Gaussian kernel $\hat{x}(t) = g(0, \alpha\sigma) * x(t)$, where $g(0, \alpha\sigma)$ is a Gaussian filter of variance $\alpha\sigma$. σ represents the assumed signal echo-width, and α is a task dependent hyper-parameter defining the width of the smoothed ground truth, set in our experiments to 0.2. This idea is similar to the generation of "heatmaps" as regression targets in computer vision tasks [28,29]. Exact echo locations can still be extracted from the network prediction by locating the highest prediction peaks.

Table 1
Echo pattern parameter distributions in the simulated datasets.

Parameter\Distribution	None	Narrow	Wide
Echo width σ (time-samples)	$\sigma_0 = 9$	[9, 10.8]	[9, 13]
Center frequency ω (MHz)	$\omega_0 = 5.4$	[4, 5.4]	[3, 5.4]

4. Experimental setup

Experiments were carried on simulated data and two real phantoms. Detection accuracy was first measured on simulated data, where data can be generated flexibly with exact ground truth. The method was then applied to the layer thickness estimation task in two real-world ultrasonic scans of physical phantoms, specifically designed for tests of this nature.

Data Simulation. The training set consisted of 10,000 simulated signal, divided 80%–20% between the train and validation sets. Each signal contains 500 time-samples. Three types of training sets were generated. The first set is drawn without parameter randomization of σ, ω according to the randomization procedure proposed in Section 3.1.2. The second set was drawn with narrow distribution of σ, ω and the third with a wide distribution. The pulse width was randomized from σ_0 to $1.2\sigma_0$ for the narrow distribution and $1.4\sigma_0$ for the wide distribution. The center frequency was randomized according to the -3 dB bandwidth of the transducer for the narrow distribution and -6 dB bandwidth for the wide distribution. Table 1 shows the parameters of the three scenarios. In all experiments, the maximal number of echoes K was set to 8 and the minimal echo amplitude parameter α was 0.1.

Network Training. Training was done with the ADAM gradient-based optimization method [30] for 80 epochs with a learning rate of 0.01 and batch size of 150. Learning rate was reduced by a factor of 2 if the validation error did not improve for 12 epochs and training proceeded until validation error did not improve for 20 epochs. The network was implemented in Pytorch [31], and trained for about 2 minutes on an NVidia GeForce TitanX GPU.

Competition. In all experiments, the network is compared to standard SSR methods used for solving the echo detection task — OMP [7], SMP [8] and ISTA [9]. To handle variations in the pulse phase, dictionary atoms were constructed based on a complex variant of Eq. (5), [8]. Dictionary parameters were determined differently for each of the 3 data distributions described in Table 1. The number of pulse width and center frequency values represented in the dictionary were set to 1, 3, and 6 for the no randomization, randomization with narrow distribution, and randomization with wide distribution respectively, as more values are required to cover the wider echo distributions. For all distributions, the number of TOF values represented in the dictionary are equal to the number of sampling times $T_s = 500$. The dictionary included the Cartesian product of pulse width, center frequency and TOF values, so it contains 500, 4500 and 18,000 values for None, Narrow and Wide distributions.

Stopping conditions were optimized separately for simulation and real phantom experiments, to provide each method with the best obtainable results. In simulation, where there are up to 8 echoes in a signal, OMP and SMP ran until the residue energy is below 3% of the measured signal energy with a hard limit of 15 iterations. On the real phantoms, OMP and SMP methods ran until the residue energy is below 25% of original energy with a maximum of 4 iterations. This early stopping criterion is used in order to reconstruct only significant echoes. ISTA runs for 3 iterations with a threshold coefficient of 0.1.

Post Processing. Simple post-processing is introduced to reduce noise and promote sparsity in predictions. First, activations with values lower than a pre-defined value β are set to zero. For the physical phantoms, we set the threshold β rather high in order to catch the first two echoes. If there are less than two values above β, it is iteratively reduced by assigning $\beta = c \cdot \beta$ with $c \in [0, 1]$, until two echoes are detected. In simulation, β is set to 0.01, while with the real phantoms

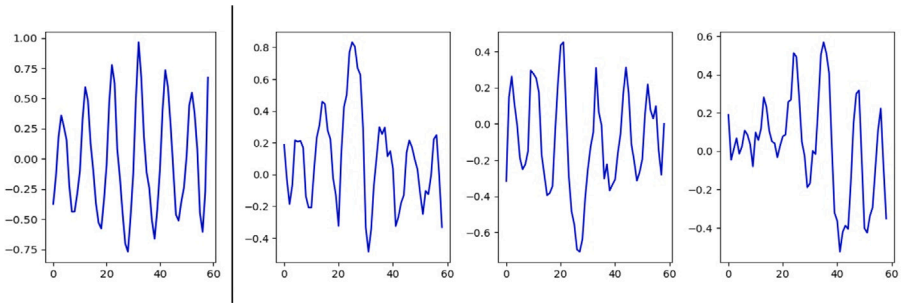

Fig. 4. Representative filters learned by the first layer of the network. Left: Filter of a network trained with one filter in the first layer. **Right:** 3 Representative filters of a network trained with 30 filters in the first layer.

it is set to 0.2 and $c = 0.8$. After the thresholding operation, non-maximum suppression (NMS) is applied to remove small detection activations which are near larger activations. The NMS range used is 14 time samples from the selected detection, which is approximately equal to $1.5\sigma_0$.

5. Results

5.1. Simulation results

Evaluation metric. To evaluate accuracy in simulation, each detection (an output activation larger than zero) is compared with all ground truth delta-functions of the signal, to see if a hit can be found. Denote a detection hypothesis and a ground truth delta by $z(t) = a_z\delta(t-t_z)$ and $x(t) = a_x\delta(t-t_x)$ respectively, where a_z, a_x are the amplitudes and t_z, t_x are the delta function timings. A detection is declared a 'hit' if

$$\frac{\sum_{t=1}^{T}(s(z(t)) - s(z(t)))^2}{\sum_{t=1}^{T}(s(x(t)))^2} < \epsilon, \tag{7}$$

where $s(\cdot)$ returns the Gaussian envelope component of the simulated signal, as described by Eq. (5). Based on this hit definition, a recall–precision curve over the test set is plotted and the area under the curve (AUC) and maximal F1 score (across all threshold values) are reported as the accuracy indices. Note that a detection is classified as a hit only if it matches a true echo in both location and amplitude. For small values of ϵ, this can be rather challenging.

Results. We report accuracy on the narrowly distributed (scenario 2 in Table 1) test set as a function of three main difficulty parameters: λ (Fig. 5 (left)) controlling echo overlap, required detection accuracy ϵ (Fig. 5 (middle)), and SNR (Fig. 5 (right)), defined in Section 3.1.2. The default values of λ, ϵ and SNR in these experiments are set to 3, 0.1, and 20 respectively, and in each experiment a single parameter changes while the other two are kept fixed. For each algorithmic method, variants corresponding to the 3 distributions (see Table 1) were tested and results are shown for the best option (narrow distribution for US-CNN, non-randomized dictionaries for the other methods).

As can be seen in Fig. 5, the network provides higher accuracy then competing algorithms in most of the scenarios. The left graph shows that the network has a considerable advantage over competition for scenarios with significant echo overlap (λ values of 1 and 2). For high overlap of $\lambda = 1$ it provides almost twice the accuracy than alternative algorithms. The middle graph shows that the network has a significant advantage when high detection accuracy is required according to Eq. (7) (small ϵ values). SMP, a greedy search algorithm tailored specifically for the task of echo separation, provides similar results to the network's in easier cases. When accuracy as a function of SNR is considered (Fig. 5 (right)), the network and SMP are the leading algorithms.

5.2. Real phantom results

We assess the network on scans taken from two physical phantoms in the task of layer thickness estimation. The first phantom is composed of two disks made from Ultem (a type of machinable polymer) attached by an adhesive layer (see Fig. 6(a)). The top Ultem disk is flat and the other has inner circular steps of 0.1 mm in height. The two disks are attached with adhesive, forming an adhesive layer of thickness ranging from 0.1 mm to 0.5 mm at intervals of 0.1 mm, according to the steps in the lower Ultem disk. The second phantom is a thin 8-step Aluminum calibration block, with thickness ranging from 0.03 inch to 0.1 inch at intervals of 0.01 inch (see Fig. 6(b)). Both phantoms were scanned by a scanning acoustic microscope equipped with a 5 MHz, flat, 0.5 inch diameter transducer.

The two phantoms differ in material and structure, posing a different echo separation challenge. First, the change in material and experimentation time leads to differences in the reflected echoes in terms of the echo patterns. Additionally, the aluminum block introduces reverberating multi reflections that result in additional overlaps after the second echo.

Thickness estimation results. Layer thickness estimations were obtained by extracting the TOFD between the two largest echoes and converting it from time (time-samples in microseconds) to distance (mm) based on the given material sound-velocity of the two materials. The inspected phantoms were premeditatedly designed and manufactured with a high level of precision for experimentation. Therefore, the exact distance in each thickness is reliably known. Fig. 7 shows the mean absolute error (MAE) obtained by all methods for the thinnest layer, and for the average of the remaining layers in the two phantoms. The thinnest layer provides information regarding the range resolution of each method, whereas the average of the remaining layers measures the overall detection accuracy and consistency. For each algorithmic method, variants corresponding to the 3 distributions (see Table 1) were tested and results are shown for the best option (narrow distribution for US-CNN, non-randomized dictionaries for OMP and SMP, and wide distribution for ISTA).

For the thinnest layers in each model, where the neighboring echoes severely overlap, US-CNN provides the best accuracy on both models, and with a significant margin. For the Ultem phantom, the network also shows the lowest error rate overall across the different layers, whereas for the Aluminum phantom ISTA provides the lower error rates. However, for the Aluminum phantom all methods successfully separated the overlapping echoes across all layers but the thinnest. This can be seen by the low MAE values, ranging from 0.015 mm to 0.034 mm, comparing to the block thicknesses, ranging from 1 mm to 2.54 mm, representing a maximal deviation of 3% from the correct value.

Thickness estimation visualization. Fig. 8 shows the detection results of the tested algorithms as two dimensional thickness maps. While all the methods perform reasonably well qualitatively for thick layers, the network provides a more accurate and stable prediction. For

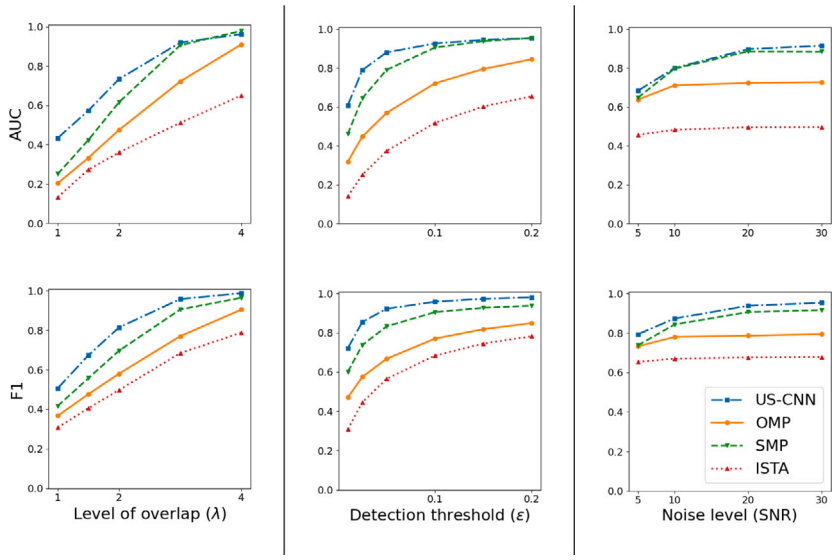

Fig. 5. Simulation results. Comparison of detection accuracy as measured by AUC (top) and F1 (bottom) among the four methods tested. Detection accuracy is plotted as a function of echo overlap parameter λ (right), required accuracy threshold ϵ (middle) and SNR (left).

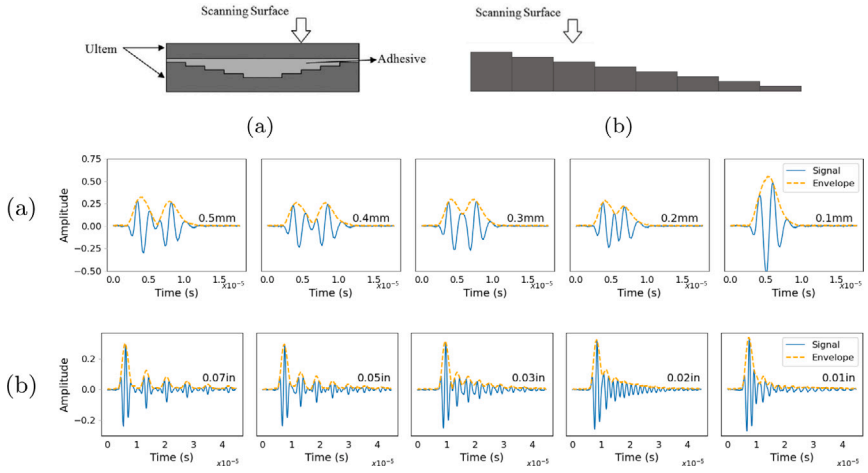

Fig. 6. Demonstration of real phantoms used for thickness estimation experimentation. Top: Side view illustrations of the phantoms: (a) Round Ultem phantom, and (b) Aluminum phantom. **Bottom:** Time-domain signals from different layers of the (a) Ultem and (b) Aluminum phantoms.

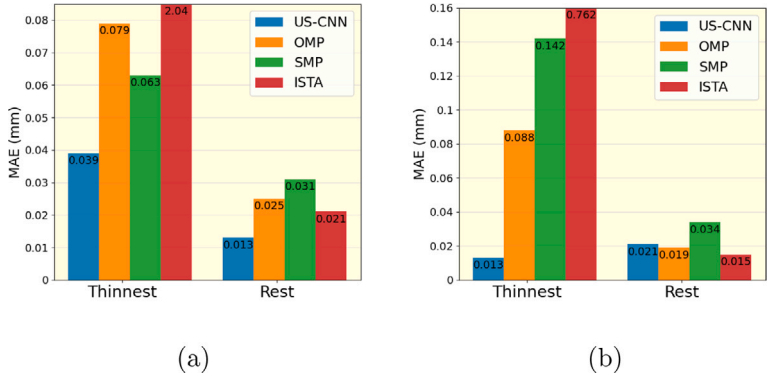

Fig. 7. Thickness estimation Results. Mean Absolute Error (MAE, in mm) of the tested methods on the (a) Ultem and (b) Aluminum phantoms. In each figure, the left histogram shows average error on the thinnest layer, the right histogram shows the average error of the remaining layers.

the thinnest layers, the network shows significant improvement over competing algorithms, as can be clearly seen by the TOFD error figures. US-CNN successfully separates overlapping echoes in the thinnest layer of both phantoms, enabling a new range resolution for layer thickness estimation. The fine differences between the methods for thin layers can be better seen in a closer inspection of the A-scans. In Fig. 9, A-scans and predictions of the tested algorithms on the round model are shown. On the left an a-scan from the 0.5 mm ring of the phantom

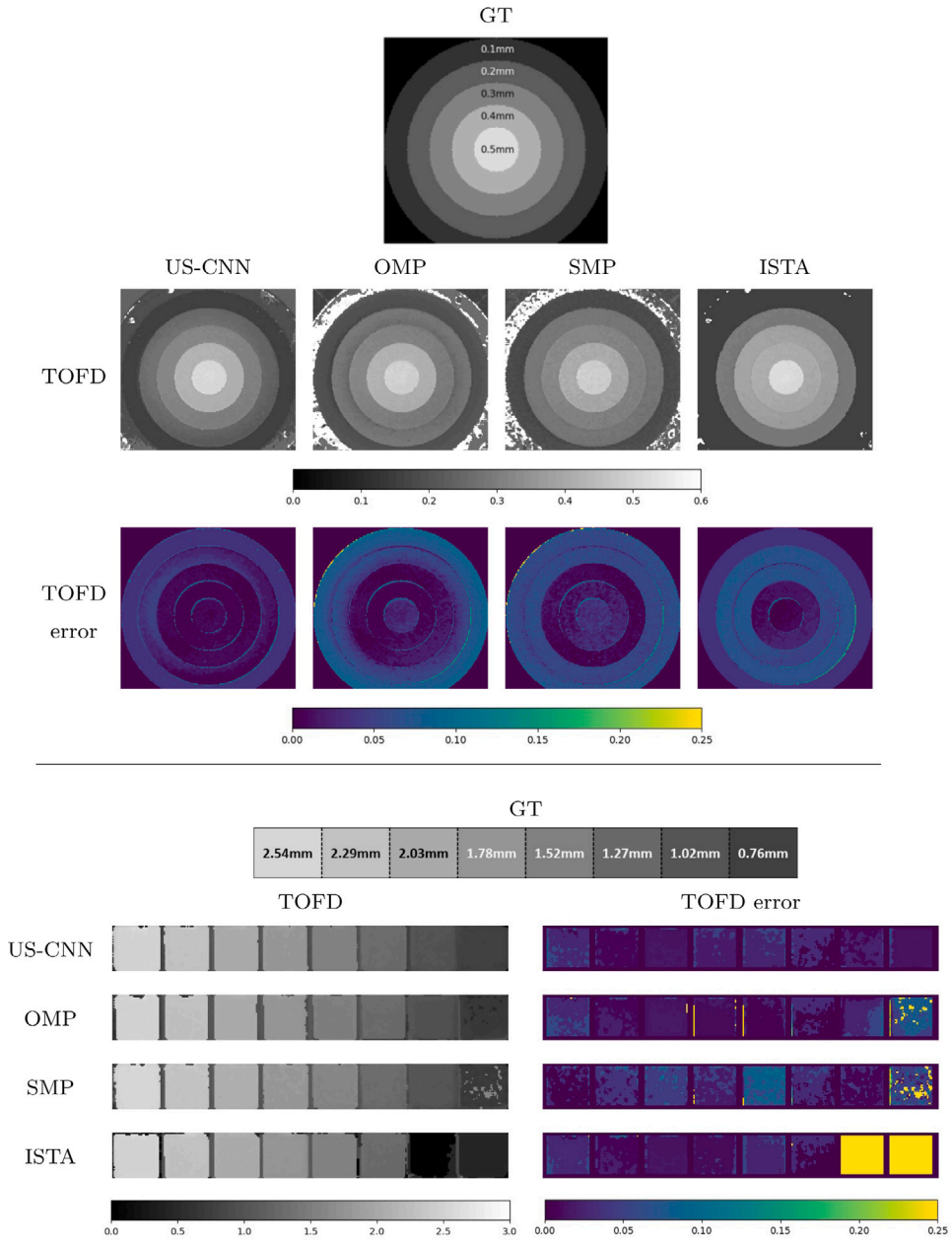

Fig. 8. Thickness estimation visualization for layer physical phantoms. Each pixel within the Time-of-Flight-Difference (TOFD) predictions (Gray scale images) indicates the estimated thickness for the spatial location. The color images are the corresponding TOFD error between the prediction and the ground truth. **Top:** Ultem phantom. **Bottom:** Aluminum phantom.

is shown, along with US-CNN prediction and the signal envelope. As can be seen, the prediction is located at the center of the signal envelope. On the right, prediction of an a-scan from the 0.1 mm ring is shown for US-CNN, SMP, and ISTA. As can be seen, SMP and the ISTA methods misplace a single large echo in the center the overlapping echoes' combination. This main central middle detection is followed by additional one or more detections corresponding to small residual tails. These methods hence fail to detect the accurate interface positions.

Amenability to signal variation. In Fig. 10, layer thickness MAE of the tested method is shown as a function of training distribution width. Results are shown separately for overall estimation (where the mean is over all estimations in all layer thicknesses) and for the thinnest layer in each model. For SSR methods, usually wider dictionaries used in wider distributions effectively causes a deterioration in thickness estimation. The network exhibits another behavior. For overall performance, the network seems to monotonically gain from training on a wider distribution. For the thinnest layers with the highest overlap, the widest distribution causes damage, but a distribution with moderate variance provides the best accuracy. These results suggest that the network is a more plausible alternative when large echo variance is present.

Inference efficiency. Network inference for a single simulated signal, containing 500 time samples, takes 0.5 ms on GPU and 1.5 ms on CPU. Processing the signal in an unoptimized manner with ISTA, OMP, and SMP, used with the minimal number of dictionary atoms, takes 5, 10 and 20 ms respectively on CPU. That is, US-CNN is 3.5–40 times faster than traditional algorithms.

6. Conclusions

In this paper a lean and efficient neural network is presented for the task of overlapping echo detection. sparse approximation techniques

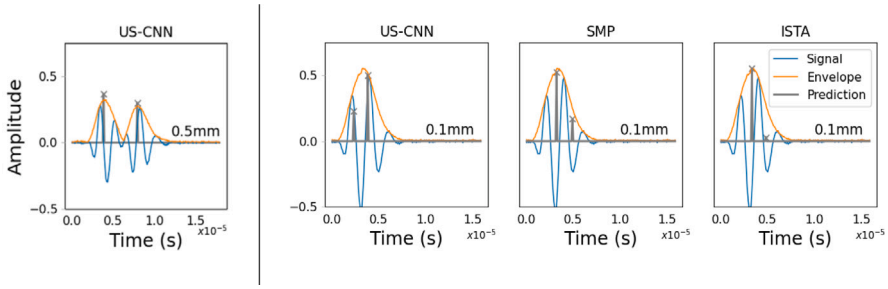

Fig. 9. A-scan Prediction visualization. Left: A-scan with US-CNN echo detection from the 0.5 mm ring of the round Ultem phantom. **Right:** A-scan from the 0.1 mm ring of the round Ultem along with detections of US-CNN, SMP and ISTA.

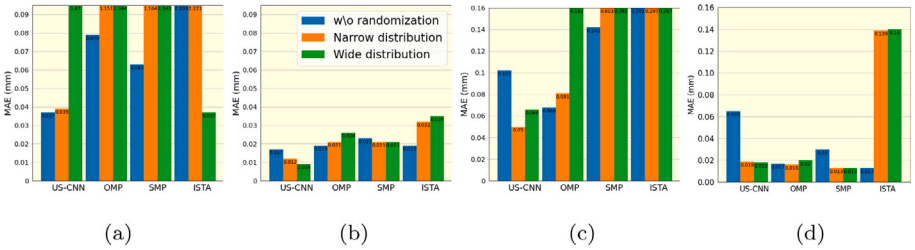

Fig. 10. Error as a function of training distribution width. Mean absolute errors of the four tested methods as a function of training distribution width. **(a)–(b) Round phantom.** MAE of (a) thinnest layer and (b) average of the rest of the layers. **(c)–(d) Rectangular phantom.** MAE of (c) thinnest layer and (d) average of the rest of the layers. Very large errors (full height bars) are measured when a method fails to find a second echo relevant for width estimation.

in the task of overlapping echo detection. Using simulated data, the network was shown to provide better accuracy than competition in high overlap conditions and when high detection accuracy is required. Tested on two physical phantoms, the network provided with accurate and stable estimation in a layer thickness estimation task. Experiments with increased echo variance in the training phase showed that compared to dictionary-based models, the network provides a more robust alternative for coping with high echo variance. In further work, the network model can be adapted to additional signal physical characteristics, such as chirps and multi-reflections, simply by adding those to the training data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the Pazy foundation Israel, through grant no. ID63-2018.

References

[1] G.-M. Zhang, C.-Z. Zhang, D.M. Harvey, Sparse signal representation and its applications in ultrasonic NDE, Ultrasonics 52 (3) (2012) 351–363.
[2] T. Pialucha, P. Cawley, The detection of thin embedded layers using normal incidence ultrasound, Ultrasonics 32 (6) (1994) 431–440.
[3] L. Brekhovskikh, Waves in Layered Media, vol. 16, Elsevier, 2012.
[4] R. Demirli, J. Saniie, Model-based estimation of ultrasonic echoes. Part I: Analysis and algorithms, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 48 (3) (2001) 787–802.
[5] R. Demirli, J. Saniie, Model-based estimation pursuit for sparse decomposition of ultrasonic echoes, IET Signal Process. 6 (4) (2012) 313–325.
[6] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[7] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inform. Theory 53 (12) (2007) 4655–4666.
[8] E. Mor, A. Azoulay, M. Aladjem, A matching pursuit method for approximating overlapping ultrasonic echoes, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 57 (9) (2010) 1996–2004.
[9] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math. 57 (11) (2004) 1413–1457.
[10] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[11] A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, 2016, arXiv preprint arXiv:1609.03499.
[12] J.-P. Briot, G. Hadjeres, F.-D. Pachet, Deep learning techniques for music generation–a survey, 2017, arXiv preprint arXiv:1709.01620.
[13] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, T. Sainath, Deep learning for audio signal processing, IEEE J. Sel. Top. Sign. Proces. 13 (2) (2019) 206–219.
[14] Y. Xu, J. Du, L.-R. Dai, C.-H. Lee, A regression approach to speech enhancement based on deep neural networks, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (1) (2014) 7–19.
[15] S. Pascual, A. Bonafonte, J. Serra, SEGAN: Speech enhancement generative adversarial network, 2017, arXiv preprint arXiv:1703.09452.
[16] D. Rethage, J. Pons, X. Serra, A wavenet for speech denoising, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2018, pp. 5069–5073.
[17] Y. Konforti, A. Shpigler, B. Lerner, A. Bar-Hillel, Inference graphs for CNN interpretation, in: European Conference on Computer Vision, Springer, 2020, pp. 69–84.
[18] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, Omnipress, 2010, pp. 399–406.
[19] V. Monga, Y. Li, Y.C. Eldar, Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing, 2019, arXiv preprint arXiv:1912.10557.
[20] Z. Li, T. Wu, W. Zhang, X. Gao, Z. Yao, Y. Li, Y. Shi, A study on determining time-of-flight difference of overlapping ultrasonic signal: Wave-transform network, Sensors 20 (18) (2020) 5140.
[21] K.F. Kaaresen, Deconvolution of sparse spike trains by iterated window maximization, IEEE Trans. Signal Process. 45 (5) (1997) 1173–1183.
[22] M.S. O'Brien, A.N. Sinclair, S.M. Kramer, Recovery of a sparse spike time series by L/sub 1/norm deconvolution, IEEE Trans. Signal Process. 42 (12) (1994) 3353–3365.
[23] R. Demirli, J. Saniie, A generic parametric model for ultrasonic signal analysis, in: 2009 IEEE International Ultrasonics Symposium, IEEE, 2009, pp. 1522–1525.
[24] F. Boß mann, G. Plonka, T. Peter, O. Nemitz, T. Schmitte, Sparse deconvolution methods for ultrasonic NDT, J. Nondestruct. Eval. 31 (3) (2012) 225–244.
[25] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[26] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[28] V. Lempitsky, A. Zisserman, Learning to count objects in images, Adv. Neural Inf. Process. Syst. 23 (2010) 1324–1332.
[29] Y. Itzhaky, G. Farjon, F. Khoroshevsky, A. Shpigler, A. Bar-Hillel, Leaf counting: Multiple scale regression and detection using deep CNNs, in: BMVC, 2018, p. 328.
[30] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8026–8037.