



# Phased microphone array for sound source localization with deep learning

Wei Ma<sup>1</sup> · Xun Liu<sup>2</sup>

Received: 8 March 2019 / Revised: 28 April 2019 / Accepted: 3 May 2019 / Published online: 14 May 2019  
© Shanghai Jiao Tong University 2019

## Abstract

To phased microphone array for sound source localization, algorithm with both high computational efficiency and high precision is a persistent pursuit until now. In this paper, convolutional neural network (CNN) a kind of deep learning is preliminarily applied as a new algorithm. The input of CNN is only cross-spectral matrix, while the output of CNN is source distribution. With regard to computing speed in applications, CNN once trained is as fast as conventional beamforming, and is significantly faster than the most famous deconvolution algorithm DAMAS. With regard to measurement accuracy in applications, at high frequency, CNN can reconstruct the sound localizations with up to 100% test accuracy, although sidelobes may appear in some situations. In addition, CNN has a spatial resolution nearly as that of DAMAS and better than that of the conventional beamforming. CNN test accuracy decreases with frequency decreasing; however, in most incorrect samples, CNN results are not far away from the correct results. This exciting result means that CNN perfectly finds source distribution directly from cross-spectral matrix without given propagation function and microphone positions in advance, and thus, CNN deserves to be further explored as a new algorithm.

**Keywords** Microphone arrays · Beamforming · Deep learning · CNN

## 1 Introduction

In recent years with the development of society, the awareness of the impact of noise on health has increased significantly, environmental comfort has been becoming more and more important, and consequently, acoustic source localization has been increasingly critical in noise diagnosis. Nowadays, phased microphone array has become a standard technique for acoustic source localization. In the post-processing, the main two categories of traditional algorithms are beamforming and deconvolution algorithms.

Beamforming algorithms construct a dirty map of source distributions from array microphone pressure signals [1].

---

Wei Ma and Xun Liu contributed equally to this work.

✉ Wei Ma  
mawei@sjtu.edu.cn

Xun Liu  
ae1905kaka@gmail.com

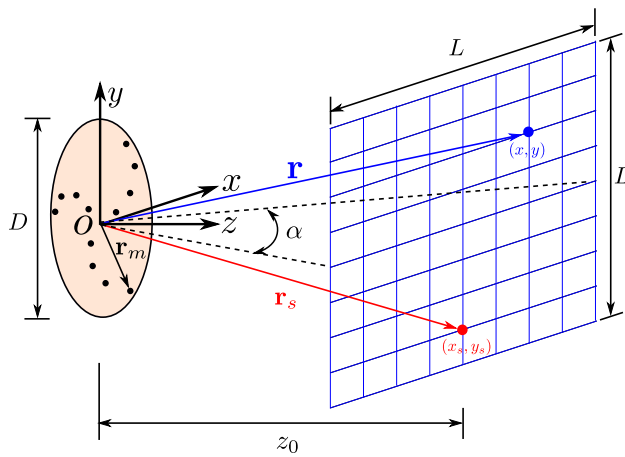
<sup>1</sup> School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, People's Republic of China

<sup>2</sup> Shanghai KeyGo Technology Company Limited, Shanghai, People's Republic of China

Conventional beamforming is simple and robust; however, its main disadvantages include poor spatial resolution particularly at low frequencies and poor dynamic range due to side-lobe effects [2]. For algorithms with better performances, many researchers have proposed some advance beamforming algorithms, such as orthogonal beamforming [3], robust adaptive beamforming [4], and functional beamforming [5]. Concerning spatial resolution, these advance beamforming algorithms have obvious superiority compared to conventional beamforming; however, they are not as good as deconvolution algorithms.

Deconvolution algorithms reconstruct a clean map of source distributions from a dirty map via iterative deconvolution, and thus can significantly improve the spatial resolution. The most famous deconvolution algorithms are DAMAS [6,7], NNLS [8], and CLEAN-SC [9]. However, deconvolution algorithms require a relatively high computational effort compared to conventional beamforming due to the inevitable iterations used in the deconvolution algorithms. Spectral procedure [10] and compression computational grid [11–13] are used to improve the efficiency of deconvolution algorithms.

There are still two big challenges for phased microphone array. One is that algorithm with both high computational



**Fig. 1** Sketch of setup with a phased microphone array and a two-dimensional region of interest. Origin of the coordinate system is placed in the centre of the microphone array

efficiency and high precision is a persistent pursuit, to improve the ability of real-time display and online analysis. The other one is that when phased microphone array used in complex flow environment with unknown propagation function, phased microphone array with traditional algorithms loses its accuracy, due to uncertainty in the propagation function used in traditional algorithms when.

At this time, deep learning—deep neural networks—is the most attractive data-mining tool without any doubt. Deep learning is a specific kind of machine learning [14]. Machine learning is able to learn from data and find the relationship between input and output data. Deep learning discovers intricate structure in large data sets using the back propagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer [15]. Deep learning has recently achieved spectacular success in many domains such as speech recognition [16], visual object recognition [17], astronomy [18], as well as the game of Go [19].

In traditional disciplines, deep learning has also attracted widespread attention and expected to be able to further solve the traditional problems. For example, deep learning has been used to turbulence modelling in fluid mechanics [20,21]. In these traditional disciplines, deep learning is still strongly challenging the deep-rooted consensus that innovations are inspired from expert-in-the-loop intuition and physically interpretable models, by providing competing predictions and without clear physical interpretation.

Inspired by the success of deep learning, in this paper, convolutional neural network (CNN) [14] a kind of deep learning is applied to phased microphone array for sound source localization as a new algorithm. CNN uses the mathematical operation convolution in at least one of their layers.

Convolution leverages three important ideas that can help improve a machine learning: sparse interactions, parameter sharing, and equivariant representations [14].

This attempt mainly looks forward to making full use of three features of deep learning to overcome the big challenges of phased microphone array introduced above. The first one is the excellent data-learning capabilities. The second one is its computational speed once trained. The last one is its potential applications with unknown propagation function and microphone positions.

The rest of this paper is organized as follows. Algorithms are presented in Sect. 2. An application is examined in Sect. 3. A discussion is presented in Sect. 4. Finally, conclusions are given in Sect. 5.

## 2 Algorithms

Figure 1 illustrates a setup with a phased microphone array that contains  $M$  microphones and has a diameter of  $D$ , as well as a two-dimensional region of interest. Stationary noise sources are located in an  $x$ - $y$  plane at a distance of  $z_0$  from the centre of the microphone array. The length of the scanning plane is  $L = 2z_0 \tan(\alpha/2)$ , where  $\alpha$  is the opening angle. The region of interest is divided into  $S = N \times N$  equidistant points.

In each test case, data from the microphone array are simultaneously acquired. Cross-spectral matrix (CSM) is then calculated using these simultaneously acquired data from the microphone array. The acquired data of each microphone are divided into  $I$  frames. Each frame is then converted into frequency bins by Fast Fourier Transform (FFT). For a given angular frequency  $\omega$ , CSM is averaged over  $I$  blocks:

$$\mathbf{C}(\omega) = \overline{\mathbf{p}(\omega)\mathbf{p}(\omega)^H} = \frac{1}{I} \sum_{i=1}^I \mathbf{p}_i(\omega)\mathbf{p}_i(\omega)^H, \quad (1)$$

where  $\mathbf{p}(\omega) = [p_1(\omega), p_2(\omega), \dots, p_M(\omega)]^T$ , and  $(\cdot)^H$  denotes complex conjugate transpose. For the sake of brevity,  $\omega$  is omitted in the following. The problem of phased microphone arrays for source localization can be expressed as

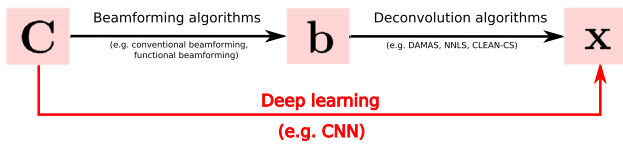
$$f(\mathbf{C}) = \mathbf{x}, \quad (2)$$

where  $\mathbf{x}$  is the source distribution of power descriptors and

$$\mathbf{x} = [|q_1|^2, \dots, |q_s|^2, \dots, |q_S|^2]^T, \quad (3)$$

where  $q_s$  is source amplitude in terms of the pressure produced at source point  $s$ .

Figure 2 shows algorithms dealing with Eq. 2 for phased microphone array, including beamforming algorithms, decon-



**Fig. 2** Algorithms for phased microphone array. *C* CSM, *b* beamforming, *x* source distribution

volution algorithms, and deep learning. Beamforming algorithms obtain beamforming from CSM, and deconvolution algorithms obtain source distribution from beamforming, while deep learning proposed in this paper obtain source distribution directly from CSM.

### 2.1 Beamforming algorithm

The conventional beamforming

$$b(\mathbf{r}) = \frac{\mathbf{e}(\mathbf{r})^H \mathbf{C} \mathbf{e}(\mathbf{r})}{\|\mathbf{e}(\mathbf{r})\|^4}, \tag{4}$$

where the vector  $\mathbf{e}(\mathbf{r}) \in \mathbb{C}^{M \times 1}$  is the steering vector at  $\mathbf{r}$  and

$$\mathbf{e}(\mathbf{r}) = [e_1(\mathbf{r}), \dots, e_m(\mathbf{r}), \dots, e_m(\mathbf{r})]^T. \tag{5}$$

The notation of steering vector under monopole point source assumption and in a medium with a uniform flow is [7]

$$e_m(\mathbf{r}) = \frac{\|\mathbf{r} - \mathbf{r}_m\|}{\|\mathbf{r}\|} \exp\{-j2\pi f/c_0\|\mathbf{r} - \mathbf{r}_m\|\}, \tag{6}$$

where  $\|\mathbf{r}\|$  is the distance from a beamformer focus position to the centre of the array,  $\|\mathbf{r} - \mathbf{r}_m\|$  is the distance from the beamformer focus position to the *m*th microphone (see in Fig. 1), *f* is frequency, and *c*<sub>0</sub> is speed of sound.

### 2.2 Deconvolution algorithms

The sound pressure contribution at microphones can be written as

$$\mathbf{p} = \sum_{s=1}^S \mathbf{e}(\mathbf{r}_s) q_s. \tag{7}$$

For incoherent acoustic sources, CSM thus becomes

$$\mathbf{C} = \sum_{s=1}^S |q_s|^2 \mathbf{e}(\mathbf{r}_s) \mathbf{e}(\mathbf{r}_s)^H. \tag{8}$$

The conventional DAS beamforming output can then be written as

$$\begin{aligned} b(\mathbf{r}) &= \sum_{s=1}^S |q_s|^2 \cdot \frac{\mathbf{e}(\mathbf{r})^H [\mathbf{e}(\mathbf{r}_s) \mathbf{e}(\mathbf{r}_s)^H] \mathbf{e}(\mathbf{r})}{\|\mathbf{e}(\mathbf{r})\|^4} \\ &= \sum_{s=1}^S |q_s|^2 \cdot \frac{|\mathbf{e}(\mathbf{r})^H \mathbf{e}(\mathbf{r}_s)|^2}{\|\mathbf{e}(\mathbf{r})\|^4}. \end{aligned} \tag{9}$$

For a single unit-power point source, Eq. (9) is known as point-spread function (PSF) of the array and is defined as

$$\text{PSF}(\mathbf{r}|\mathbf{r}_s) = \frac{\mathbf{e}(\mathbf{r})^H [\mathbf{e}(\mathbf{r}_s) \mathbf{e}(\mathbf{r}_s)^H] \mathbf{e}(\mathbf{r})}{\|\mathbf{e}(\mathbf{r})\|^4} = \frac{|\mathbf{e}(\mathbf{r})^H \mathbf{e}(\mathbf{r}_s)|^2}{\|\mathbf{e}(\mathbf{r})\|^4}, \tag{10}$$

and then, Eq. (9) can then be written as

$$b(\mathbf{r}) = \sum_{s=1}^S |q_s|^2 \cdot \text{PSF}(\mathbf{r}|\mathbf{r}_s). \tag{11}$$

By computing PSF( $\mathbf{r}|\mathbf{r}_s$ ) for all combinations of ( $\mathbf{r}|\mathbf{r}_s$ ) in discrete grid and arranging each resulting PSF map columnwise in a matrix **A**, Eq. (11) could reformulate in matrix notation as

$$\mathbf{A} \mathbf{x} = \mathbf{b}, \tag{12}$$

where **b** contains the beamformer map. Equation (12) is a system of linear equations. Notice that  $\mathbf{A} \in \mathbb{R}^{S \times S}$ ,  $\mathbf{x} \in \mathbb{R}^{S \times 1}$ ,  $\mathbf{b} \in \mathbb{R}^{S \times 1}$ .

The deconvolution task is to find a source distribution **x** for a give dirty map **b** and know matrix **A**. The constraint is that each component of the vector **x** is larger or equal to zero. In most of the applications, the matrix **A** is singular, and **b** is in the range of **A**, and this means that there are very large number of solutions of **x** that fulfil Eq. 12.

The DAMAS algorithm [7] is an iterative algebraic deconvolution method. In this algorithm, the source distribution is calculated by the solution of Eq. 12 using a Gauss–Seidel-type relaxation. In each step, the constraint is applied that the source strength remains positive.

### 2.3 Deep learning

CNN, a kind of deep learning, is used in this paper to phased microphone array for sound source localization as a new algorithm. This attempt mainly looks forward to making full use of prominent features of deep learning to overcome the big challenges of phased microphone array. These prominent features mainly include its excellent data-learning capabilities of

and its computational speed once trained. Keras framework [22] with a Tensorflow backend is used here.

In this paper, the input and output tensors of CNN are  $\mathbf{C} \in \mathbb{C}^{M \times M}$  and  $\mathbf{x} \in \mathbb{R}^{S \times 1}$ , respectively. This makes CNN as two distinct differences with beamforming algorithms and deconvolution algorithms. The first difference is that CNN does not need in advance the propagation equation, which is a prerequisite in beamforming algorithms and deconvolution algorithms. The second difference is that CNN even does not need in advance the positions of microphones in phased microphone array, which is also a prerequisite in beamforming algorithms and deconvolution algorithms. The first difference makes that CNN has a significant advantage that CNN can be used in lots applications, where propagation equation is unknown. The second difference makes that CNN has another advantage that CNN can avoid the errors caused by the position deviations of microphones in installation. Of course, the prerequisite for these advantages is that CNN can find  $\mathbf{x}$  directly from  $\mathbf{C}$  with high accuracy.

In this subsection, CNN settings including network architecture, training data, and training strategy are introduced. In the next section, applications are carried out to check the ability of CNN.

### 2.3.1 Networks' architecture

A CNN model with seven layers is proposed, as depicted in Fig. 3. The parameters and structures of this CNN model are listed in Table 1. This CNN model consists of four two-dimensional convolutional layers (Conv2D), two two-dimensional pooling layers (MaxPooling2D), a flatten layer (Flatten), and a regular densely connected neural networks layer (Dense). The convolutional layers perform discrete convolution operations on their input. In each convolutional layer, zero-padding is valid, such that the output has the same length as the original input; meanwhile, a bias vector is created and added to the outputs. The output of each convolutional layer is passed to a rectified linear unit (ReLU) filter. The pooling layer performs a max operation over sub-regions of the extracted feature maps resulting in down sampling by a factor of two. The flatten layer just flattens the input and does not affect the batch size. The regular densely connected neural network layer gives  $S$ -dimensional output space using a matrix multiplication and bias addition.

The number of trainable parameters mainly depends on microphone number  $M$  and grid number  $S$ . For example, in the applications in the next section, CNN has approximately  $7.23 \times 10^5$  trainable parameters with microphone number  $M = 30$  and grid number  $S = 100$ .

### 2.3.2 Training data

The data used to train the network are collected through simulation. In this process, different sound source distributions  $\hat{\mathbf{x}}$  are assigned in advance, and then, corresponding  $\mathbf{C}$  are calculated according to Eq. 8.

During the training process, 80%, 10%, and 10% of all data generated are selected as the training data, validation data, and test data, respectively. To avoid the network simply memorizing the training data rather than learning general features for accurate prediction with new data, validation data should not appear in training data. We deleted the samples in validation data that already exist in training data. In addition, we deleted the samples in test data that already exist in training data or in validation data.

### 2.3.3 Training strategy

The loss function used to train the weights of the networks is set as the mean squares of the errors between assigned and predicted values for a sample, such as

$$\text{Loss} = \frac{1}{S} \sum_{i=1}^S (\hat{x}_i - x_i)^2 = \frac{1}{S} \sum_{i=1}^S \left( |\hat{q}_i|^2 - |q_i|^2 \right)^2, \quad (13)$$

where  $\hat{x}_i$  and  $x_i$  are assigned and predicted source powers at  $i$ th grid, respectively, and  $\hat{q}_i$  and  $q_i$  are assigned and predicted source pressure at  $i$ th grid, respectively.

In the training, the Adam stochastic optimization algorithm is used to update the network parameters with learning rate of 0.001 and mini-batch size of 32 samples. The number of epochs to train the model is specified as 20, which appeared to be more than enough for convergence. The network training takes around some hours on a MacBook Pro with a processor of 2.9 GHz Inter Core i5.

The accuracy is defined as

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{validation}}} \times 100\%, \quad (14)$$

where  $N_{\text{validation}}$  is number of samples in validation data and  $N_{\text{correct}}$  is number of identified correctly. Theoretical criterion for a sample is identified correctly is that  $\mathbf{x} = \hat{\mathbf{x}}$  for this sample. However, this theoretical criterion is too strict.

In the next section for sampling training data, the given sound distributions are set as **equal power** sound sources randomly distributed in the grid. In this paper, a simple criterion for a sample is identified correctly and is used. For a sample with  $N_s$  assigned sound sources, a sequence of predicted sources is first arranged by their power magnitude from large to small. Subsequently, the locations of the top  $N_s$  sources in this sequence constitute a set  $\{y_1, \dots, y_{N_s}\}$ . This sample

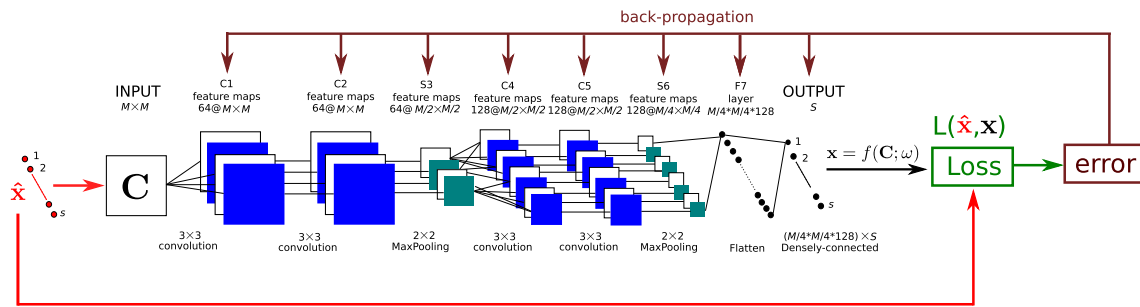


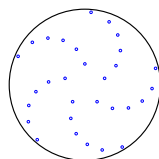
Fig. 3 Architecture of the CNN model

Table 1 Parameters and structures of the CNN model

Layer no.	Layer type	Kernel number	Kernel size	Stride	Activation	Padding	Output size
1	Conv2D	64	$3 \times 3$	$1 \times 1$	ReLU	Yes	$M \times M \times 64$
2	Conv2D	64	$3 \times 3$	$1 \times 1$	ReLU	Yes	$M \times M \times 64$
3	MaxPooling2D	–	$2 \times 2$	$2 \times 2$	–	No	$\frac{M}{2} \times \frac{M}{2} \times 64$
4	Conv2D	128	$3 \times 3$	$1 \times 1$	ReLU	Yes	$\frac{M}{2} \times \frac{M}{2} \times 128$
5	Conv2D	128	$3 \times 3$	$1 \times 1$	ReLU	Yes	$\frac{M}{2} \times \frac{M}{2} \times 128$
6	MaxPooling2D	–	$2 \times 2$	$2 \times 2$	–	No	$\frac{M}{4} \times \frac{M}{4} \times 128$
7	Flatten	–	–	–	–	–	$\left(\frac{M}{4} \times \frac{M}{4} \times 128\right) \times 1$
8	Dense	1	$S \times \left(\frac{M}{4} \times \frac{M}{4} \times 128\right)$	–	–	–	$S \times 1$

$S$  grid number,  $M$  microphone number

Fig. 4 Phased microphone array, 30-channel irregular microphones, and diameter of 0.35 m



is identified correctly only if

$$\{y_1, \dots, y_{N_s}\} = \{\hat{y}_1, \dots, \hat{y}_{N_s}\}, \tag{15}$$

where  $\{\hat{y}_1, \dots, \hat{y}_{N_s}\}$  is the set of locations of assigned sources.

### 3 Applications

In this section, synthetic applications are carried out to check the ability of CNN for sound source localization.

A planar array that contains 30 simulated microphones and has a diameter  $D$  of 0.35 m, as shown in Fig. 4, is used in these applications. In the geometrical setup, the observation plane is parallel to the array plane, and the region of interest is right in front of the array. The distance between array plane and

observation plane  $z_0$  is 2.0 m. The opening angle  $\alpha = 45^\circ$ . The computational grid is  $10 \times 10$  with 100 grid points.

In this section, conventional beamforming and DAMAS are used as traditional beamforming algorithm and deconvolution algorithm, respectively. For these two traditional algorithms, diagonal removal is applied on the CSM used for conventional beamforming, while no diagonal removal is applied on the PSF used for DAMAS. DAMAS is run with 1000 iterations, which appeared to be more than enough for convergence.

CNN, the new algorithm in this paper and has described in the previous section, has approximately  $7.23 \times 10^5$  trainable parameters according to microphone number  $M = 30$  and grid number  $S = 100$ . For sampling training data, the given sound distributions are set as **up to six** equal power sound sources randomly distributed in the grid.

#### 3.1 Effect of number of training data

In the training process, CNN are fed with training data. CNN test accuracy is affected by the number of training data. With too little training data, CNN cannot fully learn the laws hidden in the data, and then, its prediction accuracy will be



**Table 2** Accuracy of CNN with number of training data,  $f = 5$  kHz

Number of training data (k)	Accuracy (%)
5	68.4 ± 4.5
10	80.8 ± 2.5
20	88.0 ± 1.7
50	93.7 ± 0.9
90	95.6 ± 0.5
100	100.0 ± 0.0

**Table 3** Accuracy of CNN with frequency, 100 k training data

$f$ (kHz)	Accuracy (%)	$R/\Delta x$
5	100.0±0.0	3.27
4	90.0±1.5	4.08
3	70.8±3.8	5.44

greatly discounted. With too much training data, the training process needs huge time and thus is harmful to industrial applications of CNN, although CNN can fully learn the laws hidden in the data. For sound source localization, the effect of number of training data on CNN is investigated in this subsection.

CNN test accuracy with number of training data is listed in Table 2. To exclude the effect of frequency, the frequency keeps constant and is 5 kHz. The CNN test accuracy is only 68.4% with the number of training data of 5 k. With the number of training data increasing, CNN test accuracy increases. When the numbers of training data are 10 k, 20 k, 50 k, and 90 k, the CNN test accuracies are 80.0%, 88.0%, 93.7%, and 95.6%, respectively. The CNN test accuracy is up to 100% when the number of training data reaches 100 k. In addition, the uncertainty decreases when the number of training data increases.

### 3.2 Effect of frequency

For sound source localization, frequency is one of the most important factors. When frequency decreases, the spatial resolution of the conventional beamforming increases following the relationship  $R = 1.22zc/(\cos(\alpha/2)^3 Df)$ , where  $c$  is sound velocity. DAMAS as a convolution algorithm, its spatial resolution is also increases with frequency decreasing, although it is significant improved compared to the conventional beamforming. In this section, the effect of frequency on CNN is investigated.

CNN test accuracy with frequency is listed in Table 3. To exclude the effect of number of training data, the number of training data is 100 k in these trainings. In this table,  $R/\Delta x$ , where  $\Delta x$  is the space between the grid nodes, is also listed to show the development of the spatial resolution with fre-

quency. As has discussed in the previous subsection, CNN test accuracy at  $f = 5$  kHz is up to 100%. CNN test accuracy decreases, as the frequency decreases. CNN test accuracies are only 90.0% and 70.8% at  $f=4$  kHz and 3 kHz, respectively.

Four samples in test data at  $f = 5$  kHz are shown in Figs. 5, 6, 7 and 8. Reconstruction maps of the conventional beamforming, DAMAS, and CNN are shown in these figures. With regard to computing speed in applications, CNN is as fast as conventional beamforming, and is significantly faster than DAMAS. In Fig. 5, three equal power sound sources exist in the scanning plane. The distance between these points are larger than  $3.27\Delta x$ . As a consequence, these three sound sources could be distinguished in the conventional beamforming map, as shown in Fig. 5a. DAMAS reconstructs these three sound sources successfully with better spatial resolution, as shown in Fig. 5b. CNN can also reconstruct successfully these three sound sources with the same better spatial resolution as DAMAS, as shown in Fig. 5c. In Fig. 6, five equal power sound sources exist in the scanning plane. The distance between points is larger than  $3.27\Delta x$  except the two sound sources placed at grid points (6, 10) and (9, 10). As a consequence, most of these sound sources could be distinguished in the conventional beamforming map, as shown in Fig. 6a. DAMAS reconstructs these sound sources successfully with better spatial resolution, as shown in Fig. 6b. CNN reconstructs successfully these sound sources with the same better spatial resolution as DAMAS, as shown in Fig. 6c; however, the same sidelobes appear. In Fig. 7, six equal power sound sources exist in the scanning plane. The aggregation degree of these sound sources is larger than that in Fig. 6. Two sound sources are placed at adjacent grid points. As a consequence, the conventional beamforming cannot distinguish these two adjacent sound sources, as shown in Fig. 7a. DAMAS reconstructs these sound sources successfully with better spatial resolution even for these two adjacent sound sources, as shown in Fig. 7b. CNN reconstructs successfully these sound sources with the same better spatial resolution as DAMAS, as shown in Fig. 7c. However, sidelobes appear, and the number of sidelobes is more than that in Fig. 6c. In addition, the reconstruction power of the sound source placed at grid point (5, 9) is obvious smaller than others. In Fig. 8, six equal power sound sources exist in the scanning plane. The aggregation degree of these sound sources are larger than that in Fig. 7. Two sound sources are placed at adjacent grid points, and there is only one grid point between these two adjacent sources to another one. As a consequence, the conventional beamforming cannot distinguish these adjacent sound sources, as shown in Fig. 8a. DAMAS reconstructs these sound sources successfully with better spatial resolution even for these two adjacent sound sources, as shown in Fig. 8b. CNN reconstructs successfully these sound sources with the same better spatial resolution as DAMAS, as shown

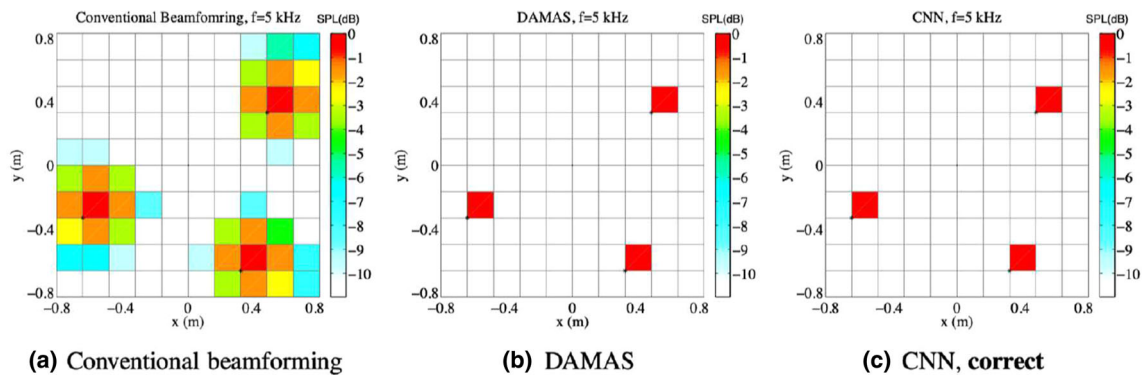


Fig. 5 Sample at  $f = 5$  kHz. Black cross symbols, positions of synthetic point sources

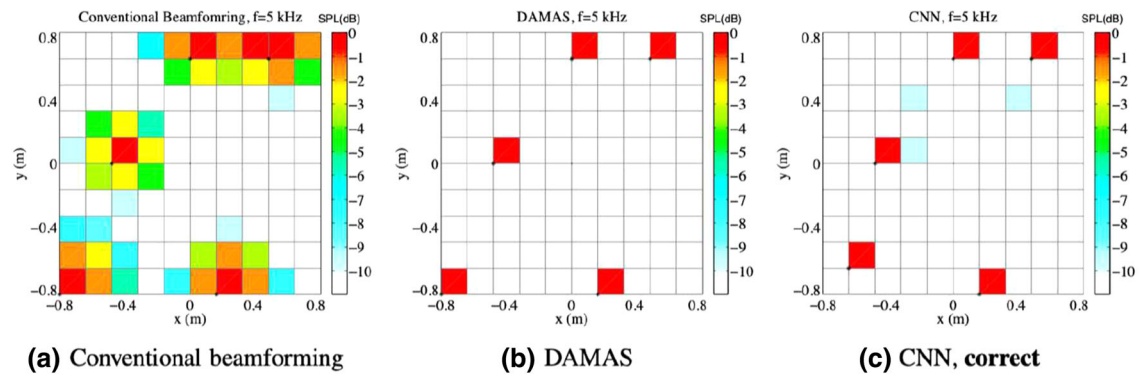


Fig. 6 Sample at  $f = 5$  kHz. Black cross symbols, positions of synthetic point sources

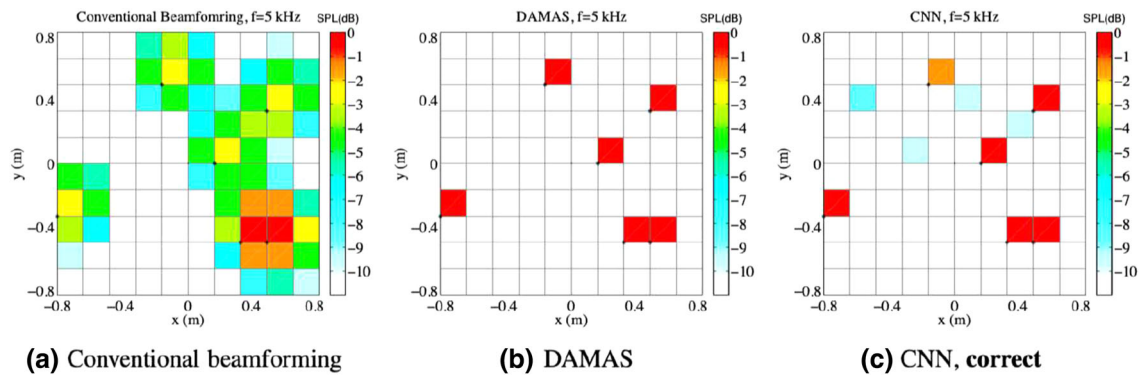


Fig. 7 Sample at  $f = 5$  kHz. Black cross symbols, positions of synthetic point sources

in Fig. 7c. However, sidelobes appear, and the number of sidelobes is more than that in 7c.

Figures 9, 10, 11 and 12 show the reconstruction results at  $f = 3$  kHz for the given sound distributions in Figs. 5, 6, 7 and 8, respectively. With the frequency decreases from 5 to 3 kHz, the spatial resolutions of the conventional beamforming and DAMAS deteriorate. Focusing on CNN, more sidelobes appears; meanwhile, power deviations appear significantly which induce CNN gives incorrect results in Fig. 12.

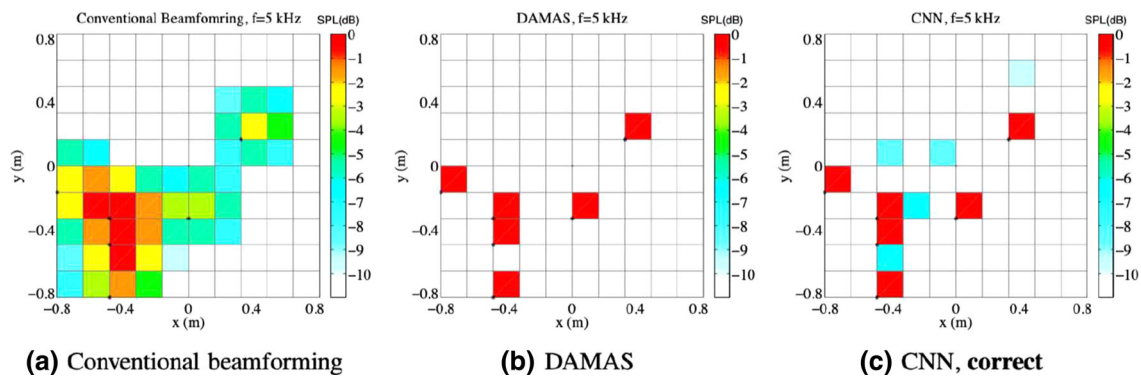
The performances of CNN in this section especially at  $f = 5$  kHz clearly indicate that CNN can be used for sound

source localization, and is an alternative full of expectations for phased microphone array.

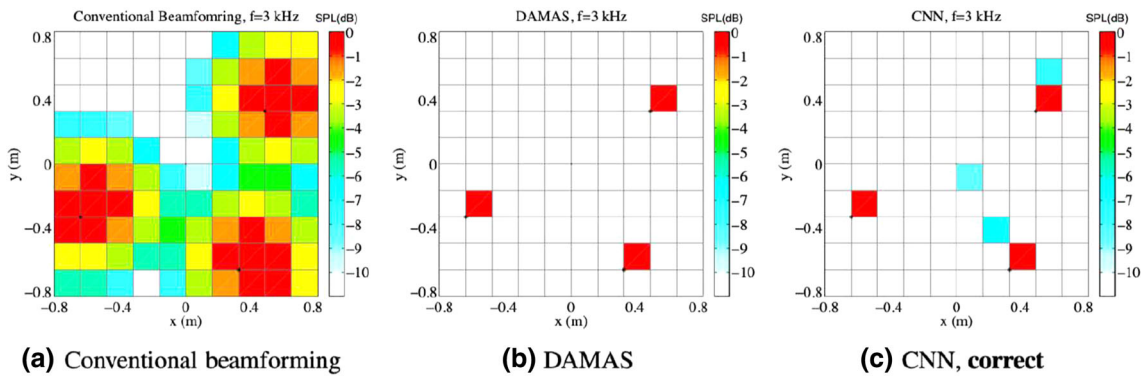
### 3.3 Incorrect samples

To further investigate the CNN performance, more incorrect samples at  $f = 3$  kHz are shown in Fig. 13.

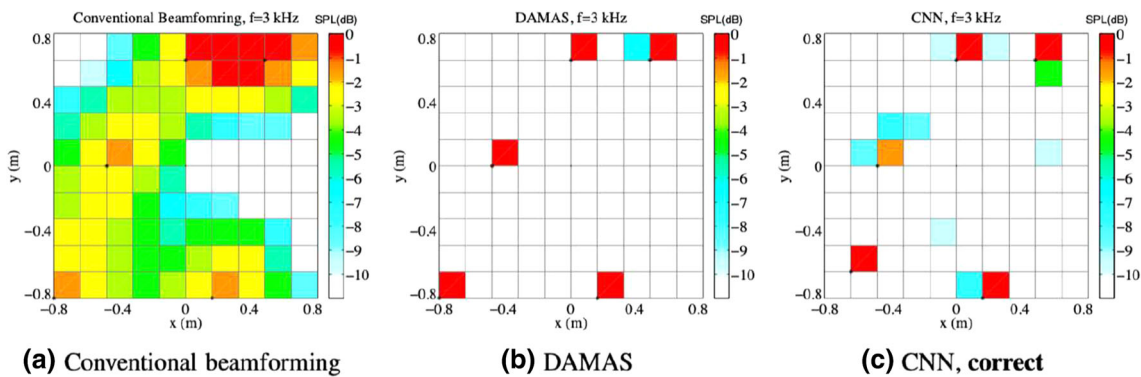
A CNN result is judged as incorrect if in this CNN result, the power at one of the assigned source points is smaller than that at a non-source grid point, according to the criterion of a sample is identified correctly (i.e., Eq. 15). This usually



**Fig. 8** Sample at  $f = 5$  kHz. Black cross symbols, positions of synthetic point sources



**Fig. 9** Sample at  $f = 3$  kHz, with the same source positions in Fig. 5. Black cross symbols, positions of synthetic point sources



**Fig. 10** Sample at  $f = 3$  kHz, with the same source positions in Fig. 6. Black cross symbols, positions of synthetic point sources

happens due to most of the power at the assigned source points moves to the adjacent grid point. Examples can be found in Fig. 13. In Fig. 13a, most of the power at the grid point (4, 10) moves to the adjacent grid point (3, 10). In Fig. 13b, most of the power at the grid point (8, 7) moves to the adjacent grid point (8, 8). In Fig. 13c, most of the power at the grid point (5, 5) moves to the adjacent grid points (6, 5) and (6, 4). In Fig. 13d, most of the power at the grid point (5, 4) moves to the adjacent grid point (5, 5). In Fig. 13e, most of the power at the grid point (5, 4) moves to the adjacent

grid point (4, 4). In Fig. 13f, most of the power at the grid point (7, 6) moves to the adjacent grid point (7, 7).

In these incorrect samples, CNN results are not far away from the correct results, due to that only same powers at assigned source points move to adjacent grid points.

## 4 Discussion

In this paper, CNN a kind of deep learning as an alternative algorithm is preliminarily applied to phased microphone



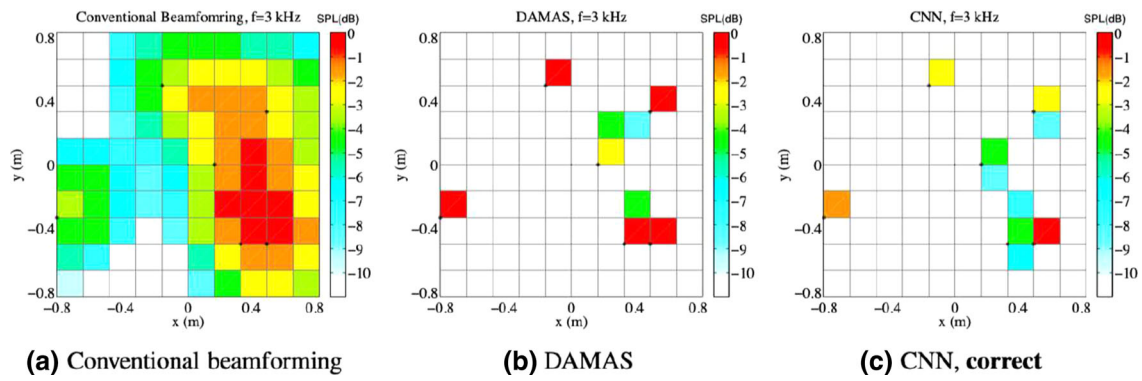


Fig. 11 Sample at  $f = 3$  kHz, with the same source positions in Fig. 7. Black cross symbols, positions of synthetic point sources

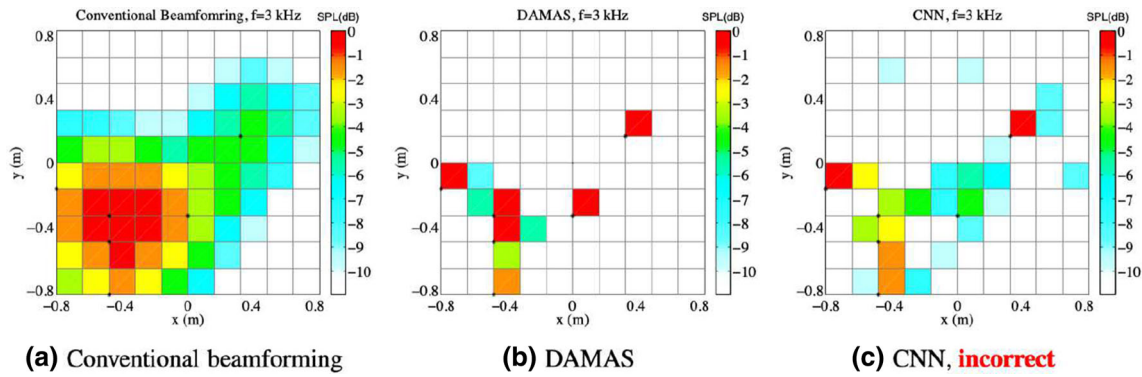


Fig. 12 Sample at  $f = 3$  kHz, with the same source positions in Fig. 8. Black cross symbols, positions of synthetic point sources

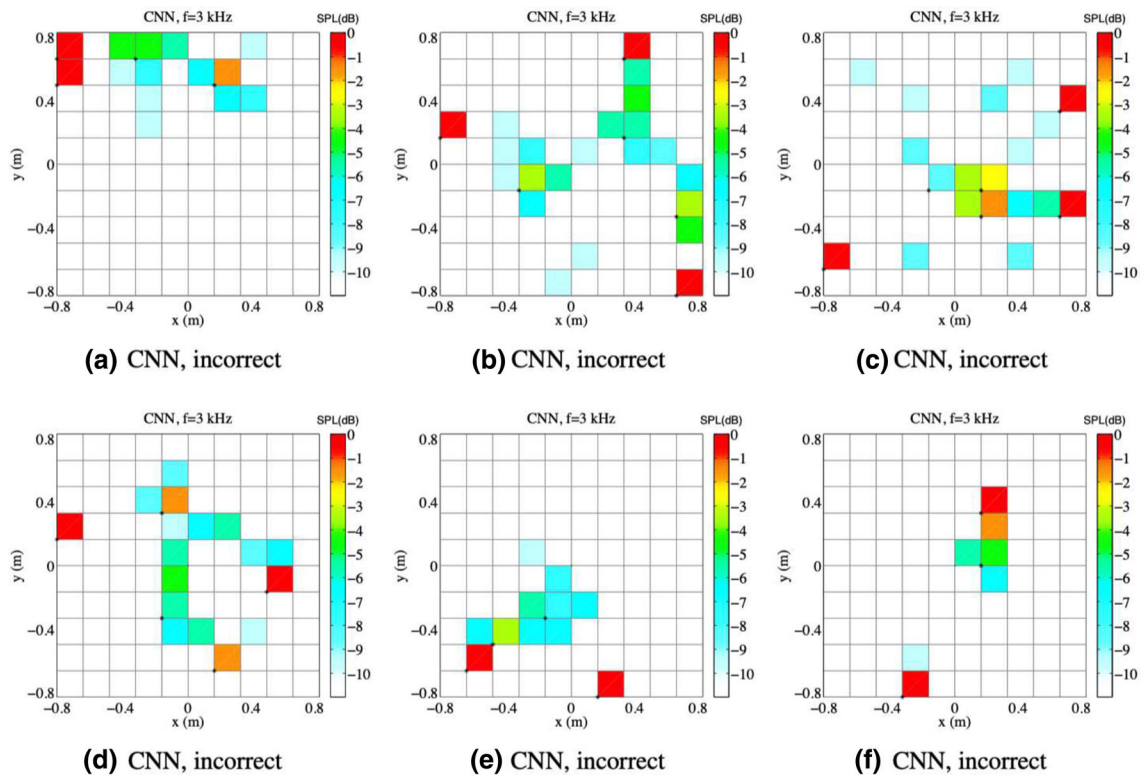


Fig. 13 Incorrect samples,  $f = 3$  kHz. Black cross symbols, positions of synthetic point sources

arrays for sound source localization. To the best knowledge of the authors, this paper is first work so far that applies deep learning to phased microphone array for sound source localization. This attempt mainly looks forward to making full use of prominent features of deep learning to overcome the big challenges of phased microphone array. These prominent features mainly include its excellent data-learning capabilities of and its computational speed once trained.

Preliminary investigations are carried out to check the performance of CNN. With regard to computing speed in applications, CNN is as fast as conventional beamforming, and is significantly faster than DAMAS. With regard to measurement accuracy in applications, at high frequency, CNN can reconstruct the sound localizations with even 100% test accuracy, although sidelobes may appear in some situations. In addition, CNN has a spatial resolution nearly as that of DAMAS and better than that of the conventional beamforming. CNN test accuracy decreases with frequency decreasing; however, in most incorrect samples, CNN results are not far away from the correct results, due to only some powers at assigned source points move to adjacent grid points. This exciting result makes that CNN has encouraging prospects as a new algorithm for phased microphone array, and deserves to be further explored.

In preliminary applications, CNN almost perfectly finds source distribution, in which the input and output tensors of CNN are  $\mathbf{C} \in \mathbb{C}^{M \times M}$  and  $\mathbf{x} \in \mathbb{R}^{S \times 1}$ , respectively. This input makes that CNN has two distinct differences with beamforming algorithms and deconvolution algorithms. The first difference is that CNN does not need in advance the propagation equation, which is a prerequisite in beamforming algorithms and deconvolution algorithms. The second difference is that CNN even does not need the positions of microphones, which is also a prerequisite in beamforming algorithms and deconvolution algorithms. The first difference leads an advantage of CNN that CNN could be used to applications with unknown propagation equation, which is often encountered in complex measurement environment. The second difference leads another advantage of CNN that CNN could avoid the errors due to installation position deviation of microphones.

In preliminary applications, in this paper, training data are obtained by simulation, and there are up to six equal sound sources randomly distributed in 100 grids. For this sound sources setting, there are  $C_{100}^1 + C_{100}^2 + C_{100}^3 + C_{100}^4 + C_{100}^5 + C_{100}^6 = 1.2715 \times 10^9$  possibilities. In the applications, 100 k training data are only about 0.08% of all the possibilities. CNN is thus definitely not data fitting. This is one point that the authors want to emphasize here. In this paper, training is conducted at a giving frequency. For speeding up in applications, we suggest that taking frequency as a training parameter in the future.

One main challenge of CNN is that large amount of training data are required. Concerning training data, there are mainly three types. The first type is simulated data as used in this paper. This type of training data is suitable for applications with known propagation function, such as sound field with stationary sound sources. The second type is simulated data from computational aeroacoustics (CAA). This type of training data is suitable for applications with unknown propagation function; however, the sound field can be simulated by CAA with certain accuracy, such as aircraft wing sound field. The third type is experimental data. This type of training data is suitable for applications, especially for sound field with complex flow field, such as sound field of aero-engine fan, where velocity gradients exist in the flow field and significantly affect the propagation function of sound. Experimental data can only be accumulated through a large number of experiments with a process that takes a lot of time and money, and thus may more “expensive” than the first and the second types of training data. To save time and money, a more appropriate strategy is using combined training data including all these three types could be used in the future. Of course, this strategy should be investigated further in advance.

Another main challenge of CNN is that large time for network training is required. For example, in this paper, the network training takes some hours on a normal personal computer. This issue can be ameliorated using GPUs, whose price is getting cheaper with the development of computer hardware and strong demand due to more and more applications of deep learning in industry.

About the CNN investigation and optimization, the questions are still open and needed to investigate in the future, such as: (i) what’s the dynamic range of CNN? (ii) How many layers are most suitable for a give data set? (iii) How many kernel number and size are needed? (iv) How big are training data? (v) What is the uncertainty of CNN predictions? (vi) How to improve the accuracy at low frequency? (vii) How to use combined training data including simulated data and experimental data?

## 5 Conclusions

In this paper, CNN a kind of deep learning as an alternative algorithm is preliminarily applied to phased microphone arrays for sound source localization. To the best knowledge of the authors, this paper is first work so far that applies deep learning to phased microphone array for sound source localization. Preliminary investigations are carried out to check the performance of CNN. With regard to computing speed in applications, CNN after training is as fast as conventional beamforming, and is significantly faster than the most famous deconvolution algorithm DAMAS. With

regard to measurement accuracy, at high frequency, CNN can reconstruct the sound localizations with up to 100% test accuracy, although sidelobes may appear in some situations. CNN test accuracy decreases with frequency decreasing. In addition, in most incorrect samples, CNN results are not far away from the correct results. This exciting result means that CNN almost perfectly finds source distribution directly from cross-spectral matrix without given propagation function and microphone positions in advance. In addition, thus, CNN deserves to be further explored as a new algorithm for sound source localization.

## References

1. Johnson DH, Dudgeon DE (1993) Array signal processing: concepts and techniques. Prentice Hall, New Jersey
2. Michel U (2006) History of acoustic beamforming. BeBeC-2006-01, 1–17
3. Sarradj E (2010) A fast signal subspace approach for the determination of absolute levels from phased microphone array measurements. *J Sound Vib* 329:1553–1569
4. Huang X, Long B, Vinogradov I, Peers E (2012) Adaptive beamforming for array signal processing in aeroacoustic measurements. *J Acoust Soc Am* 131:2152–2161
5. Dougherty RP (2014) Functional beamforming. In: 5th Berlin beamforming conference 2014, BeBeC-2014-01
6. Brooks TF, Humphreys WM (2004) A deconvolution approach for the mapping of acoustic sources (DAMAS) determined from phased microphone arrays. AIAA-2004-2954
7. Brooks TF, Humphreys WM (2006) A deconvolution approach for the mapping of acoustic sources (DAMAS) determined from phased microphone arrays. *J Sound Vib* 294:856–879
8. Lawson CL, Hanson RJ (1995) Solving least square problems (Chapter 23). SIAM,
9. Sijtsma P (2007) CLEAN based on spatial source coherence. *Int J Aeroacoust* 6:357–374
10. Dougherty RP (2005) Extension of DAMAS and benefits and limitations of deconvolution in beamforming. AIAA 2005–2961
11. Ma W, Liu X (2017) Improving the efficiency of DAMAS for sound source localization via wavelet compression computational grid. *J Sound Vib* 395:341–353
12. Ma W, Liu X (2017) DAMAS with compression computational grid for acoustic source mapping. *J Sound Vib* 410:473–484
13. Ma W, Liu X (2018) Compression computational grid based on functional beamforming for acoustic source localization. *Appl Acoust* 134:75–87
14. Goodfellow I, Bengio Y, Courville A (2017) Deep learning. [www.deeplearningbook.org](http://www.deeplearningbook.org)
15. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444
16. Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans Audio Speech Lang Process* 20:30–42
17. Krizhevsky A, Hinton SIG (2012) Imagenet classification with deep convolutional neural networks. In: Communications of the ACM 60
18. Hezaveh YD, Levasseur LP, Marshall PJ (2017) Fast automated analysis of strong gravitational lenses with convolutional neural networks. *Nature* 548:555–557
19. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez Arthur T, Hubert Baker L, Lai M, Bolton A, Chen Y, Lillcrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D (2017) Mastering the game of go without human knowledge. *Nature* 550:354–359
20. Ling J, Kurzawski A, Templeton J (2016) Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J Fluid Mech* 807:155–166
21. Kutz JN (2017) Deep learning in fluid dynamics. *J Fluid Mech* 814:1–4
22. Chollet F (2015) Keras, GitHub Repository