Journal of Experimental & Theoretical Artificial Intelligence

ISSN: 0952-813X (Print) 1362-3079 (Online) Journal homepage: https://www.tandfonline.com/loi/teta20

Deep learning-based framework for expansion, recognition and classification of underwater acoustic signal

Guanghao Jin, Fan Liu, Hao Wu & Qingzeng Song

To cite this article: Guanghao Jin, Fan Liu, Hao Wu & Qingzeng Song (2020) Deep learning-based framework for expansion, recognition and classification of underwater acoustic signal, Journal of Experimental & Theoretical Artificial Intelligence, 32:2, 205-218, DOI: 10.1080/0952813X.2019.1647560

To link to this article: https://doi.org/10.1080/0952813X.2019.1647560

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE
2020, VOL. 32, NO. 2, 205–218
https://doi.org/10.1080/0952813X.2019.1647560

Taylor & Francis
Taylor & Francis Group

ARTICLE

Deep learning-based framework for expansion, recognition and classification of underwater acoustic signal

Guanghao Jin, Fan Liu, Hao Wu and Qingzeng Song

School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin, China

ABSTRACT

Recently, deep learning has developed rapidly and contributed in many fields like the classification in radar and sonar applications. In some special fields like the underwater acoustic signals, the dataset for training may be scarce due to the reason of security or other restrictions, which affects the performance of the deep learning methods as those need a big dataset to ensure high accuracy. Furthermore, the original dataset is in some formats like audio, which makes those methods difficult to capture features, especially in insufficient sample case because of the interference. In this paper, we present a novel framework that applies the LOFAR spectrum for preprocessing to retain key features and utilises Generative Adversarial Networks (GAN) for the expansion of samples to improve the performance classification. Firstly, our framework selects proper preprocessing method based on the evaluation of the spectrum methods. Secondly, our framework revises a GAN to generate samples and built an independent classification network to ensure the quality of those. Finally, our framework applies the existing classification networks to evaluate the performance and selects the best one for real utilisation. The experimental results show that the generated samples have high quality, which can significantly improve the classification accuracy of the neural models.

ARTICLE HISTORY
Received 26 October 2018
Accepted 11 July 2019

KEYWORDS
Deep learning; Underwater acoustic signal; LOFAR spectrum; classification; GAN

Introduction

Since deep learning concept (Mohamed, Hinton, & Penn, 2012) was proposed in 2006, it has performed efficiently in classification or recognition tasks. With big dataset, it has been proved to be a better tool than the traditional machine learning technologies (Lecun, Bengio, & Hinton, 2015). For example, datasets like MNIST (Chazal, Tapson, & Schaik, 2015), CIFAR-10 (Abouelnaga et al., 2016) Abouelnaga16 and so on are numerous and various. Furthermore, those datasets can be expanded easily as it is public and accessible. On the other side, without sufficient training samples, the deep learning models may be overfitted because of the limited features that can be captured by samples.

Underwater acoustic signal (Nishida, S., Iwase, R., Kawaguchi, K., Matsuo, I., & Akamatsu, T., 2017) (Li, Y., & Zhe, C., 2017) is generally used by the operators to classify the objects under the water as Figure 1 shows. In underwater acoustic signal case, the samples may be scarce because of some reasons like the restriction of collection or security reasons, so that the number of samples becomes a bottleneck to the utilisation of deep learning methods. The expansion of samples by the geometric methods (like rotation) has a limitation as those are lack of diversity. Generative Adversarial Net (GAN)(Goodfellow et al., 2014) has been proved to be efficient to solve the shortage

CONTACT Qingzeng Song qingzengsong@tjpu.edu.cn Tianjin Polytechnic University, Binshui West Road, Tianjin, China

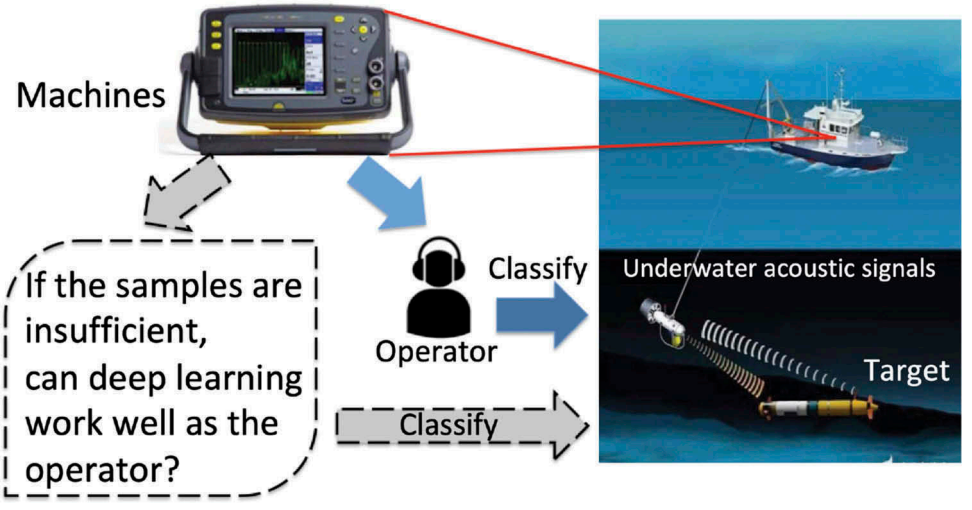

Figure 1. The underwater signal is obtained by the machines and the operator classifies the objects by the signals. If the samples are insufficient, it is a challenge to use the deep learning methods to classify the underwater acoustic signals.

of samples, which has widely used in the generation of high-quality images from original ones. Edward (Choi et al., 2017) proposed medGAN to generate dimensional different samples and Anitha (Yang, Kannan, Batra, & Parikh, 2017) presented LR-GAN (Low-Resolution GAN) to recursively generate images that have different backgrounds and foregrounds. LR-GAN is a deep neural network based on a generative adversarial network (GAN), which is to reconstruct realistic mugshot images from low-resolution probe samples.

The original samples for underwater acoustic signal are in audio format, which may contain useless information (like noise) after the translation by audio spectrum. LOFAR spectrum (Haarlem, 2016) is often utilised in underwater acoustic signal processing, which is used to track a target by experienced operators. Thus, this paper tries to use LOFAR spectrum instead of audio one for the preprocessing to reduce the difficulty of extracting features when using the deep learning methods. When using deep learning methods, the preprocessing should serve the objective of retaining key features for the following classification.

Many researchers have applied deep learning methods to classify underwater acoustic signals. In the paper [(Li, Shang, Hao, & Yang, 2016)], Xiu Li accelerated fish recognition by convolutional neural network (CNN) and achieved remarkable performance. Jonas (Jager, Wolff, Fricke-Neuderth, Mothes, & Denzler, 2017) proposed a method that has a two-stage graph with convolutional neural network (CNN) to improve the accuracy of the fish tracking. Matias (Valdenegro, 2016) used CNN for object recognition in sonar images and achieved good results. For the classification of underwater vehicles, Pingping Zhu (Zhu, Isaacs, Fu, & Ferrari, 2018) utilised CNN to classify sonar images with the technology of vector machine (SMV). In the paper (Hu et al., 2018), Gang Hu et al. proposed a method that applies CNN and ELM (extreme learning machine) to classify underwater targets, which accuracy is greatly improved compared with traditional machine learning methods with sufficient dataset. ELM is a kind of feed-forward neural network for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need not be tuned. Although those methods show that the deep learning methods can be a good tool in underwater signal classification case, those need sufficient samples to ensure high classification accuracy. Furthermore, the fitness of those networks on the generated samples by GANs also remains a problem in underwater application case.

To solve those problems, our framework firstly utilised a LOFAR spectrum translation at the preprocessing to retain key features based on our evaluation on spectrums. Then, it designed a revised GAN to expand the translated dataset to overcome the problem of insufficient samples. To ensure the generated samples to fit the deep learning methods, our framework also trained an independent classification neural network (outside the GAN) to check those samples. By those efforts, our framework can expand the samples while ensuring those can be efficiently classified by following deep learning methods. Then, some existing deep learning networks are selected to study the performance of classification on those samples. Our experimental results indicate that the generated samples can retain the key features of the targets with rich diversity and variety, so that those can be used to classification tasks by deep learning methods. Thus, our framework can be a good sample for the applications that contain the problems of insufficient samples and achieving bad performance by audio spectrum translation.

The structure of the paper is as follows: Section II introduces related work in the field of underwater acoustic signal that includes GAN and classification networks. Section III presents the materials and methods of our framework. It includes data preprocessing, the structure of our modified GAN and the classification networks that are utilised for evaluations. Section IV is the analysis of the experimental results. Finally, section V is about our conclusion and future work is also discussed.

Related work

Generative Adversarial Nets (GAN) (Gulrajani, Ahmed, Arjovsky, Dumoulin, & Courville, 2017) is recently introduced as a novel way to produce samples, which consists a model *G* to generate samples and a model *D* discriminative those. The application prospect of GAN is promising in the fields such as image generation and recognition, speech generation, natural language processing (NLP) (Yi, Nasukawa, Bunescu, & Niblack, 2003), etc. With the continuous development of GAN, many derivative models based on it were proposed. DCGAN (Radford, Metz, & Chintala, 2015) (Deep Convolutional GAN) applies a convolutional neural network to GAN to improve the stability and convergence speed of GAN. WGAN (Wasserstein GAN) (Arjovsky, Chintala, & Bottou, 2017) used Earth-Mover instead of Jensen-Shannon divergence as a training criterion to solve the problem of gradient disappearance. BEGAN (Berthelot, Schumm, & Metz, 2017) (Boundary Equilibrium GAN) proposed a new generator evaluation standard, which measures the effect of generation by estimating the difference between the data distribution error, instead of directly estimating the difference between real and fake data distribution. CGAN (Mirza & Osindero, 2014) can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information. CGAN is a relatively simple conditional generation model in the GAN derived models. ACGAN (Odena, Olah, & Shlens, 2016), InfoGAN (Chen et al., 2016) are also conditional generative models, each with its own characteristics and advantages. Among them, ACGAN is applied to assist classification, of which discriminator gives a corresponding probability for different types of input when training the generator. InfoGAN quantifies the relationship between some input and output of GAN. In that model, two additional conditional tags are given on the input random noise for training.

To evaluate the classification accuracy on expanded samples, our framework selects the existing classification networks that include LENET (Lecun, 1998), ALEXNET (Krizhevsky, Sutskever & Hinton, 2012), VGG16 (Han, Mao, & Dally, 2015), the model of Yue.H (Yue, Zhang, Wang, Wang, & Lu, 2017) and our modified network based on LENET. Those methods originally have not been used in underwater signal classification cases. Those networks are all derived from CNN. LENET consists of seven layers, consisting of two convolution layers, two pooling layers, and two fully connected layers. The convolution kernel size is 5×5, the convolution step size is 1, and the pooling is MAX. ALEXNET consists of 12 layers, five convolution layers, three pooling layers and four fully connected layers. The network uses a large convolution kernel of 11×11. There are

many versions of VGG (Han et al., 2015) and this paper uses VGG16 (Han et al., 2015) to the comparison. VGG16 is a large network with 13 convolution layers, five pooling layers and two fully connected layers.

Our framework

Structure

Figure 2 shows the whole structure of our framework and the whole process can be formulated as below:

$$\overline{S} = Gen(Pre(S)), then\ F(\overline{S}, L) = argmax_{M_i}(P(M_i(\overline{S}, L)) \tag{1}$$

where S is the set of original samples and $\overline{S}$ is the generated samples. $Pre(\cdot)$ is preprocessing by spectrums like LOFAR spectrum or audio one. The original samples are firstly transformed into audio spectrum samples and LOFAR spectrum ones. Then, our framework utilises a CNN to evaluate the two types of samples independently. Finally, our framework selects the spectrum that can achieve higher classification accuracy by CNN as the preprocessing spectrum. $Gen(\cdot)$ is to generate samples by GAN model and uses a classification network to ensure the quality of generated samples. The evaluation of the generated samples by an independent neural network is to conclude whether the generated samples can be applied to neural networks or not. Then, our framework uses existing classification network $M_i(\overline{S}, L)$ to improve the performance. Generally, a neural network $M_i(\overline{S}, L)$ can be seen as to solve the matching problem between sample $\overline{S}$ and the labels L. Then, the final objective $F(\overline{S}, L)$ can be summarised as the problem of finding the best model M_i on the generated samples for the classification. In more details, Algorithm 1 shows the process of our framework step by step.

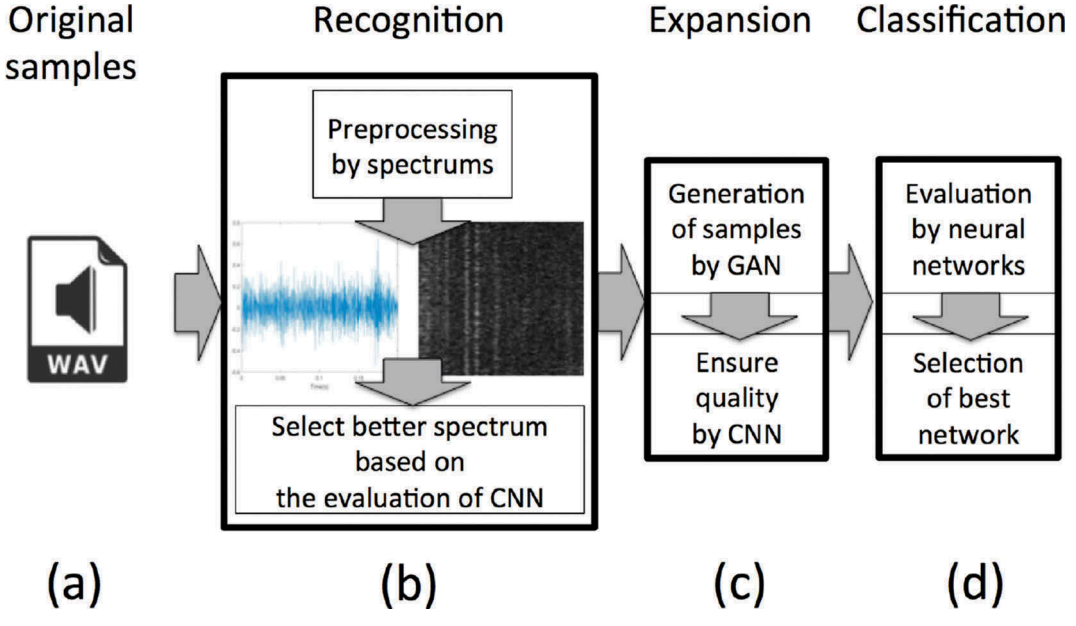

Figure 2. (a) is original samples in *.wav format. (b) is data preprocessing by spectrums and the selection of the spectrums based on a CNN, so that it ensures the samples can be recognised by neural networks. (c) is dataset expansion by GAN and an additional CNN is to ensure the quality (d) performs an evaluation on the networks and selects the best one.

Algorithm 1 The process of our framework

STEP1: Preprocessing

Set N_{sample} is the number of samples, S is the set of original samples, $N_{spectrum}$ is the number of the spectrums, $Pre^{(i)}(\cdot)$ is a preprocessing method by a spectrum, $i = (1, ..., N_{spectrum})$.

For $i = 1$

Repeat

$S^{(i)} = Pre^{(i)}(S), i = i + 1$

until $i = N_{spectrum}$

Then we can get the sets of preprocessed samples $S^{(i)} = \left\{S_j^{(i)}\right\}, i = (1, ..., N_{spectrum})$, $j = (1, ..., N_{sample})$.

Set $N_{train} < N_{sample}$, $L = \{L_j\}$ is the set of labels of S, $j = (1, ..., N_{sample})$, M_s is a neural network.

For $i = 1$

Repeat

Set $S^{(i)train} = \left\{S_j^{(i)}\right\}$, $L^{(i)train} = \{L_j\}, j = (1, ..., N_{train})$

Set $S^{(i)test} = \left\{S_j^{(i)}\right\}$, $L^{(i)test} = \{L_j\}, j = (N_{train} + 1, ..., N_{sample})$

Then train a version of M_s on $S^{(i)train}$ and $L^{(i)train}$

Set $P^{(i)} = Rate(M_s(S^{(i)test}), L^{(i)test})$ is to check the correct rate of M_s on $S^{(i)test}$.

until $i = N_{spectrum}$

Then we can get the better spectrum method by $Pre^* = argmax_{Pre^{(k)}}(P^{(k)})$ and corresponding set of samples $S^*, k = (1, ..., N_{spectrum})$

STEP2: Generating samples

Set $L = \{L_j\}$ is the set of labels of S^*, $j = (1, ..., N_{sample})$, $Gen(\cdot)$ is a GAN to generate samples and it is trained on S^* and L, $N_{final} > N_{sample}$.

For $i = N_{sample}$

repeat

$(S_i^*, L_i) = Gen(S^*, L, i), \ i = i + 1$

until $i = N_{final}$

Then combine the original samples and the generated samples, we can get the set of samples $\{S_i^*\}$ and the set of labels $\{L_i\}$,$i = (1, ..., N_{final})$.

STEP3: Ensure the quality of the generated samples

Set $N_{train} < N_{final}$, M_g is a neural network.

Set $S^{train} = \{S_i^*\}, L^{train} = \{L_i\}, i = (1, ..., N_{train})$

Set $S^{test} = \{S_i^*\}, L^{test} = \{L_i\}, i = (N_{train} + 1, ..., N_{final})$

Then train the neural network M_g on S^{train} and L^{train}

Set $P_g = Rate(M_g(S^{test}), L^{test})$, which is to check the correct rate of M_g on S^{test}.

If $R <$ MDT(manually defined threshold), then it goes to STEP2 to retrain $Gen(\cdot)$ by changing structure or parameters.

STEP4: Training classification networks

Set M_k is a neural network, $k = (1, ..., N_{model})$, N_{model} is the number of neural networks.

Set $N_{train} < N_{final}$, $S^{train} = \{S_i^*\}, L^{train} = \{L_i\}, i = (1, ..., N_{train})$

Then train each neural network M_k on S^{train} and $L^{train}, k = (1, ..., N_{model})$

STEP5: Testing models

Set $S^{test} = \{S_i^*\}, L^{test} = \{L_i\}, i = (N_{train} + 1, ..., N_{final})$

Set $P_k = Rate(M_k(S^{test}), L^{test})$ which is to check the correct rate of each M_k on S^{test}, $k = (1, ..., N_{model})$.

Then we can get the best model $\overline{M} = argmax_{M_k}(P_k), k = (1, ..., N_{model})$

Data preprocessing

LOFAR spectrum is a continuous time-domain sampling of the signal. It performs a short-time Fourier transform and the obtained time-varying information. It is projected to the time and frequency planes, so that it forms a three-dimensional stereogram. It is usually applied to target recognition. In the field of underwater acoustic signal processing, the task of tracking targets is achieved by identifying the line features of LOFAR spectrum images. At the first step, this paper preprocesses the original underwater acoustic signal by converting it into an audio spectrum and LOFAR one and evaluates those to select better spectrum. When using LOFAR spectrum, the original audio files are transformed to the spectrum ones by Short-Time Fourier Transform (STFT) (Durak & Arikan, 2003) operation. Next, it selects the characteristic frequency band from the corresponding frequency spectrum and converts the transformed results into a corresponding grey-scale sample according to the data energy interpolation. Figure 3 shows a comparison between the LOFAR spectrum image with the audio spectrum one. The left column (a) is the audio spectrum sample and the right column (b) is the sample of the LOFAR spectrum one. As it shows in that figure, the number of features that are in the LOFAR spectrum image is reduced and the key features (energy lines) are emphasised. Our framework selects a CNN and trains it on audio spectrum samples and LOFAR spectrum ones independently. The accuracy of classification shows that the LOFAR one achieves higher accuracy, which will be shown in the experiment of section 4.2. Those transformed images are delivered to the following GAN to generate more samples. Then, the generated samples can be utilised to train some neural networks to classify targets. At the classification step, the rationality of selecting LOFAR spectrum can be proved by the classification accuracy. In other words, the selection of a spectrum can only be decided by the related neural networks as those are the users of the samples.

Network structure of GAN

Our framework designed a special generation network based on CGAN (Mirza & Osindero, 2014) as a sample network of GAN, whose principle is shown in Eq 2. Similar to the original GAN network, our network also contains two parts: a generative model *G* and a discriminative model *D*. The *G*

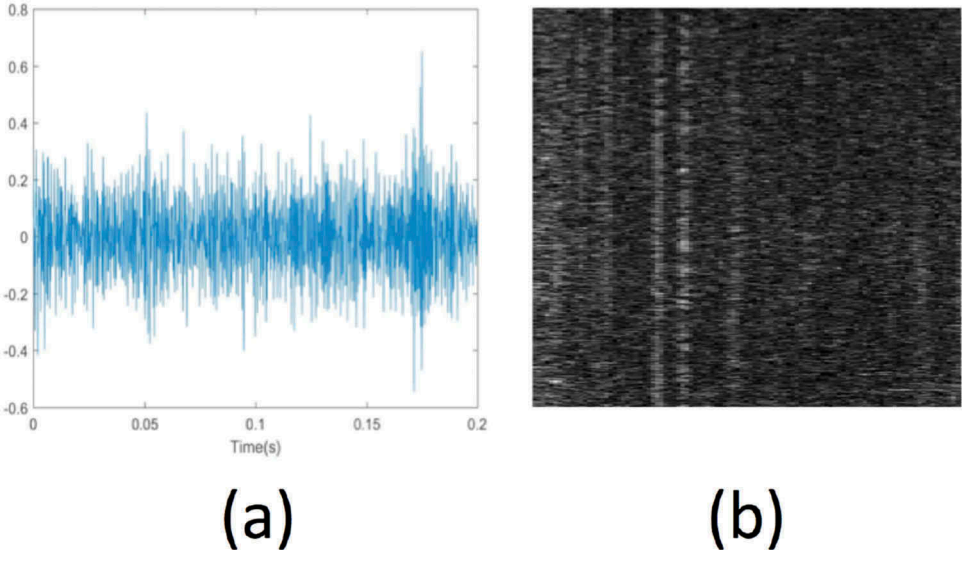

Figure 3. (a) is an audio spectrum format image (b) is a LOFAR spectrum format image.

captures the data distribution meanwhile the D estimates the probability that a sample comes from the training data rather than G. They play a dynamic game and finally reach a certain steady state that the distribution of G acquisition is as close as possible to the distribution of real samples. Differently, both of the generator and discriminator are conditioned on some extra information y. In our paper, y is the class labels and our method feeds it into the both of the discriminator and generator as an additional input layer. Figure 4 illustrates the abstracted structure of our conditional adversarial net. *Conv*1,*Conv*2 and *Conv*3 mean the convolutional layers. *FC*1 and *FC*2 mean the fully connected layers. *Upconv*1,*Upconv*2 and *Upconv*3 mean the deconvolutional layers.

$$\min_G \max_D GAN(D,G) = E_{x\sim P_{data}(x)}[logD(x|y)] + E_{z\sim p_z(z)}[log(1-D(G(z|y)))] \tag{2}$$

The generator tries to learn a distribution of generator over data x by building a mapping function from a prior noise z and its distribution $p_z(z)$ to data space as $G(z|y)$. The discriminator $D(x|y)$ outputs a single scalar representing the probability that x came from training data. We adjust parameters for G to minimise $log(1-D(G(z|y))$ and adjust parameters for D to minimise $logD(x|y)$. G and D are both trained simultaneously as if they try to achieve min-max game of function $GAN(D,G)$.

The architecture and detailed parameters of the discriminator are shown in Table 1, and the contents of generator are displayed in Table 2.

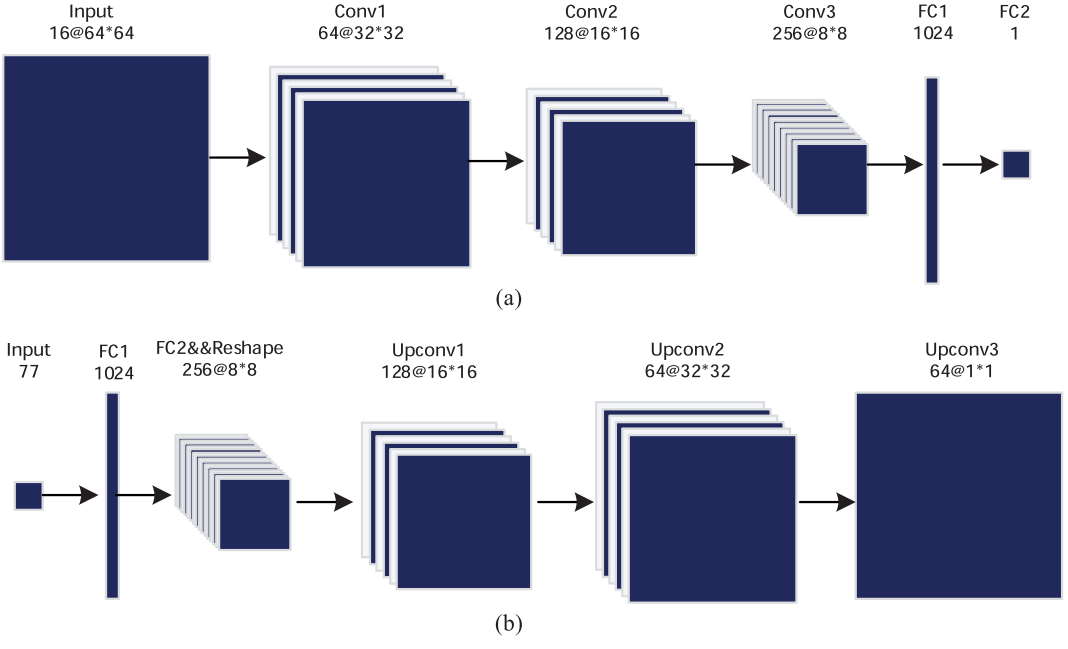

Figure 4. The overall structure of our modified CGAN.

Table 1. The design of discriminator.

Layer	Channel/Dimensions	Kernel_size	Stride	Padding	BatchNorm
Conv1	64	4×4	2	SAME	–
Conv2	128	4×4	2	SAME	BN
Conv3	256	4×4	2	SAME	BN
FC1	1024	–	–	–	BN
FC2	1	–	–	–	–

Table 2. The design of the generator.

Layer	Channel/Dimensions	Kernel_size	Stride	Padding	BatchNorm
FC1	77	–	–	–	BN
FC2	1024	–	–	–	BN
Upconv1	256	4×4	2	SAME	BN
Upconv2	128	4×4	2	SAME	BN
Upconv3	64	4×4	2	SAME	–

The principal part of the discriminator is defined by CNN, including one input layer, three convolution layers and two full-connection layers. In order to make the generated samples close to the real samples, our framework has not designed the pooling layer, because it may lose some features of the real samples. The image channel number is 16, which is the sum of the original data channel number 1 and the training data category number 15. The first convolution layer has 64 filters while the second has 128 and the third has 256, each of which has a size of 4×4 and a sliding step of 2. After three layers of convolution, it changes the dimensions of the data to two latitudes to facilitate its input as a fully connected layer, where the first dimension corresponds to the batch number and the second one corresponds to the convolved data value of a sample. After the second fully connected layer, it employs the *Sigmoid*(Kaiyrbekov, Krestinskaya, & James, 2018) activation function to map the input value between 0 and 1 in order to quantify the correctness of the sample by the distance between the value 0 and 1.

As Table 1 shows, the input layer is a four-dimensional array, in which the first three dimensions correspond to the batch size, the picture height, and the width, respectively. The fourth dimension is the sum of the original channel number of the picture and the number of conditional tag categories. In the last two convolutional layers and the first fully connected layer, it also performs LReLU (Uchida, Tanaka, & Okutomi, 2017) and batch normalisation operations. Finally, it uses the *Sigmoid* activation function to map the value between 0 and 1 to quantify the correctness of the sample.

The generator is roughly similar to that of the discriminator, but progresses in the opposite direction, including an input layer, two fully connected layers, and three de-convolution layers from front to back. The random noise vector is chosen by input layer data in our article that takes 62 as length and also adds the number of class as 15. The first de-convolution layer has 256 filters while the second one has 128 and the third one has 64 ones, each of which has a size of 4×4 and a sliding step of 2. Between the second fully connected layer and the first de-convolution layer, it performs a reshaping operation to convert the fully connected two-dimensional array into a four-dimensional array to satisfy the input data requirements of the de-convolution layer. Finally, the output processed by the generator network is a batch of images, which can be saved as a sample image after simple data processing and grey-scale mapping.

As Table 2 presents, the input layer is a two-dimensional array, the first dimension corresponds to the size of the batch, and the second dimension corresponds to the sum of the random noise vector length and the number of conditional labels which is actually a one-dimensional array of length n with a value of float, where each value is a random sample between -1 and 1. It also performs LReLU (Uchida et al., 2017) and batch normalisation operations in two fully connected layers and the first two de-convolution layers.

The structure of classification network

Our framework selects a neural network LENET as a sample to introduce and study how to revise it to improve performance. It includes an input layer, three convolutional layers, three largest pooling layers and three fully connected layers. LENET's architecture is shown in Table 3.

The modified network's architecture is shown in Figure 5 and the parameter configuration is presented in Table 4. The 64×64 LOFAR spectrum is inputted into the network. The first

Table 3. The parameters of LENET.

Name	Channel/Dimensions	Kernel_size	Stride	Padding	Type
Conv1	6	5	1	SAME	–
Pool1	6	2	2	SAME	AVG
Conv2	12	5	1	SAME	–
Pool2	12	2	2	SAME	AVG
FC1	15	–	–	–	–
FC2	15	–	–	–	–

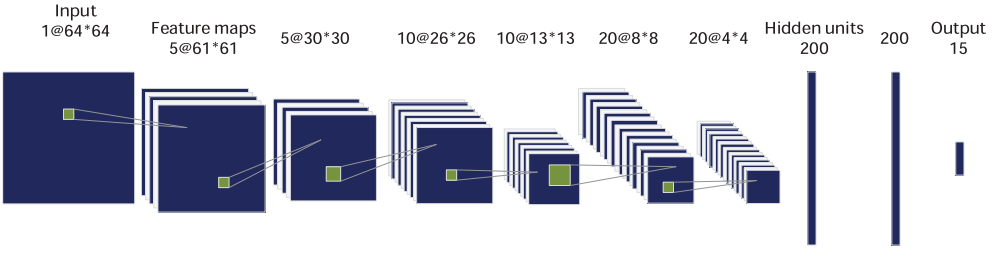

Figure 5. The architecture of modified LENET.

Table 4. The parameters of our modified LENET.

Name	Channel/Dimensions	Kernel_size	Stride	Padding	Type
Conv1	5	4	1	VALID	–
Pool1	5	2	2	VALID	MAX
Conv2	10	5	1	VALID	–
Pool2	10	2	2	VALID	MAX
Conv3	20	6	1	VALID	–
Pool3	20	2	2	VALID	MAX
FC1	200	–	–	–	–
FC2	200	–	–	–	–
FC3	15	–	–	–	–

convolutional layer has five filters, each with a size of 4×4, a sliding step size of 1 and no padding. In the pooling layers, it sets the convolution kernel size as 2×2, and the step size is 2. The purpose is to reduce the dimension of the feature map and expand the perception field. The effect of dimension reduction is to reduce the original image to a quarter and leave the strongest output. There are three combinations of such convolution and pooling layers. Each group has adjusted the size and number of convolution kernels in the convolution layer to make it more suitable for the classification to capture features of signals. More specifically, since the images are quite different, the normalisation process is performed before the first convolution operation, and the pixels of the image are scaled and shifted in formula 3, where α and β are vectors.

$$y = \alpha \times x + \beta; \frac{\partial y}{\partial x} = \alpha; \frac{\partial}{\partial \alpha} = x; \frac{\partial y}{\partial \beta} = 1 \tag{3}$$

At the end process of the network, there are three fully connected layers, and the activation function RELU (Zhang & Woodland, 2016) is applied to the first fully connected layer. In general, the output of each layer is a linear function of the input, so that there is no hidden layer. To introduce a non-linear function as an excitation function, it will no longer be a linear combination of inputs. RELU is chosen because it does not activate all neurons at the same time. If the inputted value is

less than zero, RELU will output 0, and the neuron will not be activated. The result is that only a small number of neurons are activated. The sparseness of the neural network makes it efficient and easy to calculate.

The last layer, which is the third fully connected layer, acts as a classifier. It integrates the features of convolution and pooling to capture the characteristics of the entire image and then classify the images. Finally, the output is normalised by *Softmax*(Yuan, 2017) operation, which generates a vector that presents the probabilities of the labels.

Experiment

The generated samples by our GAN

Our framework preprocessed the raw audio dataset that is obtained by real applications and labelled 15 types based on the dataset of the moving and stationary target recognition (named as MSTAR)(El-Darymli, Gill, Mcguire, & Moloney, 2017). In many of the surveyed feature-based systems, MSTAR dataset is utilised. It is a public dataset by the Sensor and Data Management System of the United States Air Force. To ensure the variance of the dataset for the classification, we expanded it by collecting the audio data of each label from the internet. According to the number of collected samples, we select the top 15 labels. Then, each sample is separated into the same time length. Then, we select 1000 samples for each label. Then, we trained our GAN on those samples and it finally generated high-quality samples. After training, our method can utilise the *G* model to generate not only a certain kind of specific spectrum samples but also various spectral ones. Some original and generated samples are shown in Figure 6, where (a) and (b) correspond to the same class while (c) and (d) correspond to various classes. As shown in Figure 6, the samples that are generated by our network contain common feature lines and all of them are clearly displayed and have more prominent characteristic lines than the original training samples.

The evaluation of spectrums and generated samples

The GAN contains a classification network, so that it ensures the classification accuracy inside the GAN. On the other side, it should prove that an independent network also could classify those generated samples. The samples produced by our approach need to be mass-produced for classifier training and 3000 samples were generated for each class. Then, it adopts AlexNet

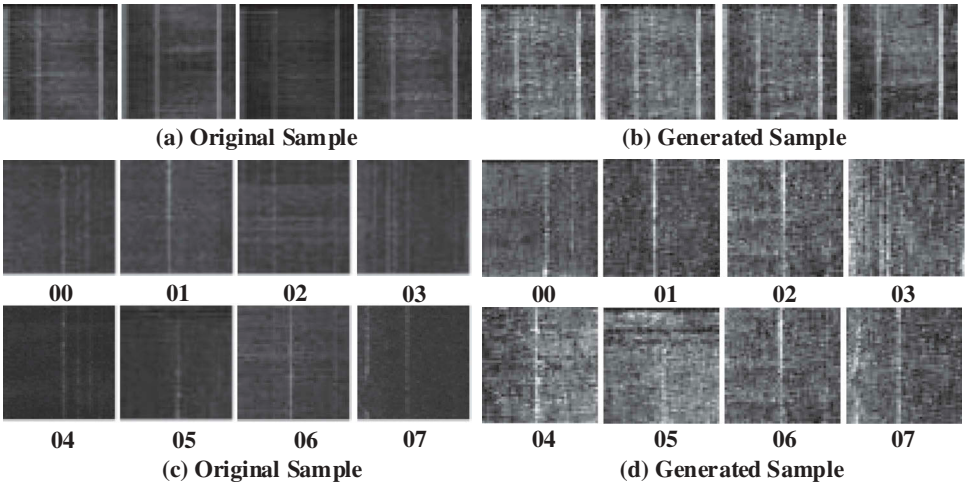

Figure 6. (a) and (c) are original samples; (b) and (d) are generated samples.

Table 5. Classification result.

Spectrum	Trainingset	Validationset	Recognition rate
Audio	original sample(80%)	original sample(20%)	52.8%
LOFAR	original sample(80%)	original sample(20%)	75.7%
LOFAR	mixed sample(80%)	mixed sample(20%)	**89.3%**

(Krizhevsky et al., 2012) as a pre-training model, and the training set contains original samples and generated samples. It mainly conducted the following two experiments:

The overall results are shown in Table 5. Firstly, our approach selects original samples and transforms those to audio spectrum ones and LOFAR spectrum ones, so that those are grouped into two datasets. Then, it separates each dataset into 80% for training and 20% for testing. The result shows that the neural network can achieve higher performance on LOFAR spectrum samples (75.7%) than the audio format ones (52.8%). Thus, our framework selects LOFAR spectrum as the preprocessing method. Secondly, it generates more LOFAR spectrum samples by our GAN. Then, it mixes original and generated samples as a whole dataset, and selects 80% for training and 20% for testing. The experiment achieves high classification accuracy (89.3%) on the mixed dataset in LOFAR spectrum format, which is higher than on the original datasets in LOFAR spectrum (75.7%) or audio spectrum (52.8%). Our experiment proves that the generated samples ensure high quality, so that those can be used to classification networks.

The evaluation of classification networks

The networks are involved in the experiment include LENET (Lecun, 1998), ALEXNET (Krizhevsky et al., 2012), VGG16 (Han et al., 2015), the model of Yue.H (Yue et al., 2017) and our modified network based on LENET. Table 6 shows the accuracy of the networks that are involved in the comparison. Our experiment also generates audio spectrum samples to compare the performance with LOFAR spectrum ones on neural networks. Our experiment compared the existing methods with our method on the audio spectrum samples and the LOFAR spectrum ones. Both of them contain the original samples and the generated samples. The experimental results show that the modified LENET (named as Our) on generated LOFAR spectrum samples has the best performance. It also shows that the classification accuracy on LOFAR spectrum samples is higher than that on audio spectrum ones. Thus, it proves the rationality of selecting LOFAR spectrum. Furthermore, it proves that the accuracy can be highly improved by expanding those samples with GAN and can be further improved by modifying the neural networks.

Discussion

Figure 7 shows the difference between audio samples and LOFAR samples on four labels. (a),(b), (c),(d) belong to the different labels. The audio samples of (a),(b) and (d) have a similar expression which causes the difficulty of classifying them in a manual way. Compare the audio sample of (a) with that of (c), it may be easy to find a difference between them in a manual way. On the other side, it is a challenge to define what are the key features that make them different from each other. Different from audio samples, the LOFAR samples of (a),(b),(c) and (d) have a different expression

Table 6. Classification results.

Network	Accuracy by Audio	Accuracy by LOFAR
LeNet	54.39%	83.64%
AlexNet	67.46%	89.30%
VGG16	67.07%	82.47%
Yue.H	32.51%	93.20%
Our	60.79%	**97.22%**

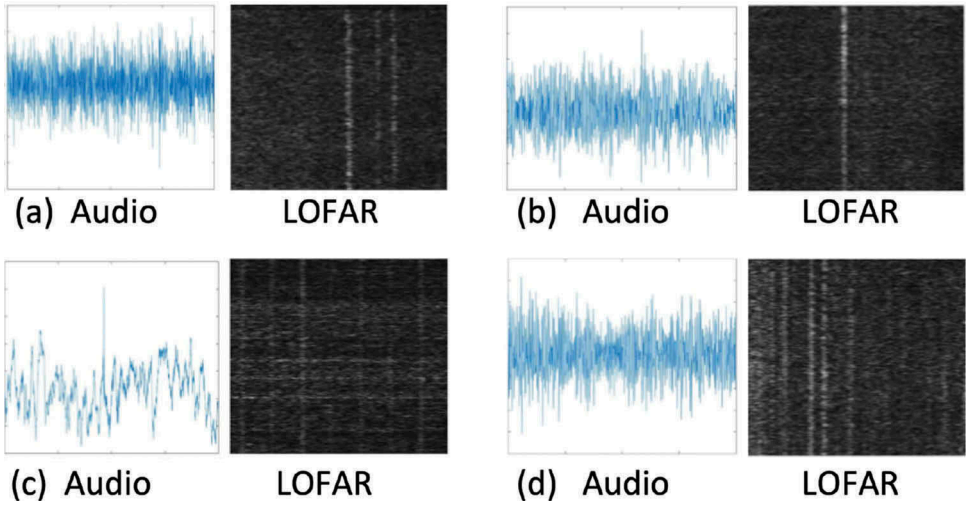

Figure 7. The difference between audio samples and LOFAR samples. (a),(b),(c),(d) belong to four different labels.

(like the location and frequency of white bands), which can be easily defined or classified by manual way or functional methods. Thus, we can draw a conclusion that it is better to use LOFAR spectrum than audio one as the preprocessing method when using deep learning methods on insufficient samples in underwater acoustic signal case.

Conclusion and future work

This paper introduced a framework that utilises a preprocessing by LOFAR spectrum for recognition and applies neural networks for expansion and classification in underwater signal processing case. Although the audio spectrum can present more information, it also contains noise that is useless for the classification of underwater signals. Thus, the deep learning networks need sufficient samples to extract the key features of objects. The LOFAR spectrum has been proved to be efficient in manual classification by the experienced operators. Thus, when the samples are insufficient, our framework utilises the LOFAR spectrum to filter noise and maintain key information for improving the efficiency of extracting features in deep learning methods. Our framework generated high-quality LOFAR spectrum samples and used those to train an independent classification network to ensure high quality. Our paper demonstrates the possibility of using GANs to generate samples that are in the LOFAR spectrum format in underwater acoustic signals. This paper also applies the existing deep learning methods on expanded samples to evaluate the performance. The classification rate of our framework can reach about 97.22%, which basically can be used for real applications. With those efforts, the experiment results show that the accuracy is highly improved in underwater acoustic signal case.

This paper shows a solution that how to solve the problem of bad performance by audio spectrum transformation or on an insufficient dataset in some applications. Furthermore, this paper shows how to modify and train generative networks like GAN to solve the insufficient dataset problem and classification networks like CNN for classification on generated samples. In the future work, we plan to study how to efficiently retain key features by other spectrums and do research about better networks to generate various samples that can simulate real images.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work has been supported by the National Natural Science Foundation of China (Grant No. 61802279, 6180021345, 61702366, 61602342 and 51607122) and Natural Science Foundation of Tianjin City [16JCYBJC41500,16JCYBJC42300,17JC-QNJC00100,18JCQNJC70300].

References

Abouelnaga, Y., Ali, O. S., Rady, H., & Moustafa, M. (2016). Cifar-10: Knn-based ensemble of classifiers. *2016 International Conference on Computational Science and Computational Intelligence. CSCI*,(pp.1192–1195). Las Vegas, NV, USA. IEEE.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan.*International Conference on Machine Learning*. Sydney, Australia.

Berthelot, D., Schumm, T., & Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks.

Chazal, P. D., Tapson, J., & Schaik, A. V. (2015). A comparison of extreme learning machines and back-propagation trained feed-forward networks processing the mnist database. *IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE*, (pp.2165–2168). Brisbane, QLD, Australia. IEEE.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., & Sun, J. (2017). Generating multi-label discrete electronic health records using generative adversarial networks.

Durak, L., & Arikan, O. (2003). Short-time fourier transform: Two fundamental properties and an optimal implementation. *IEEE Transactions on Signal Processing : a Publication of the IEEE Signal Processing Society*, *51*(5), 1231–1242.

El-Darymli, K., Gill, E. W., Mcguire, P., Power, D., & Moloney, C. (2017). Automatic target recognition in synthetic aperture radar imagery: A state-of-the-art review. *IEEE Access*, (Vol.4, pp.6014–6058).IEEE.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets. *International Conference on Neural Information Processing Systems*, (Vol.3, pp.2672–-2680). Montreal, Canada MIT Press.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans.

Haarlem, M. P. V. (2016). Lofar: The low frequency array. *Astronomy & Astrophysics*, *556*(7), 629–635.

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Fiber*, *56*(4), 3–7.

Hu, G., Wang, K., Peng, Y., Qiu, M., Shi, J., & Liu, L. (2018). Deep learning methods for underwater target feature extraction and recognition. *Computational Intelligence and Neuroscience*, *2018,(2018-3-27)*, *2018*(3), 1–10.

Jager, J., Wolff, V., Fricke-Neuderth, K., Mothes, O., & Denzler, J. (2017). Visual fish tracking: Combining a two-stage graph approach with CNN-features. *Oceans*, (pp.1–6). Aberdeen, UK. IEEE.

Kaiyrbekov, N., Krestinskaya, O., & James, A. P. (2018). *Variability analysis of memristor-based sigmoid function. 2018 International Conference on Computing and Network Communications*. Astana, Kazakhstan.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems*, (Vol.60, pp.1097–1105). Nevada, USA, Curran Associates Inc.

Lecun, Y. (1998). *Lenet-5, convolutional neural networks*. Retrieved from: yann.lecun.com.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

Li, X., Shang, M., Hao, J., & Yang, Z. (2016). Accelerating fish detection and recognition by sharing CNNs with objectness learning. Shanghai, China. *Oceans*, (pp.1–5). IEEE.

Li, Y., & Zhe, C. (2017).Entropy based underwater acoustic signal detection. *International Bhurban Conference on Applied Sciences & Technology*. Islamabad, Pakistan.IEEE.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *Computer Science*, (Vol.6, pp.2672–2680).

Mohamed, A. R., Hinton, G., & Penn, G. (2012). Understanding how deep belief networks perform acoustic modelling. *IEEE International Conference on Acoustics, Speech and Signal Processing*. Kyoto, Japan. *IEEE*,(Vol.1, pp.4273–4276). IEEE.

Nishida, S., Iwase, R., Kawaguchi, K., Matsuo, I., & Akamatsu, T. (2017). Real time detection and localization system for underwater acoustic signal with cable observatories in the west Pacific Ocean. *Techno-ocean*,(pp.544–547). Kobe, Japan. IEEE

Odena, A., Olah, C., & Shlens, J. (2016). Conditional image synthesis with auxiliary classifier gans.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations*. San Juan, Puerto Rico.

Uchida, K., Tanaka, M., & Okutomi, M. (2017). Coupled convolution layer for convolutional neural network. Cancun, Mexico. *International Conference on Pattern Recognition*, (Vol.105, pp.197). IEEE.

Valdenegro-Toro, M. (2016). Object recognition in forward-looking sonar images with convolutional neural networks. *Oceans*, (pp.1–6). Monterey, CA, USA. IEEE.

Yang, J., Kannan, A., Batra, D., & Parikh, D. (2017). LR-GAN: Layered recursive generative adversarial networks for image generation.

Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003). Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. *IEEE International Conference on Data Mining*, (pp.427–434). Melbourne, FL, USA. IEEE.

Yuan, B. (2017). Efficient hardware architecture of softmax layer in deep neural network. *System-On-Chip Conference*, (pp.323–326). Seattle, WA, USA. IEEE.

Yue, H., Zhang, L., Wang, D., Wang, Y., & Lu, Z. (2017). The classification of underwater acoustic targets based on deep learning methods. *International Conference on Control, Automation and Artificial Intelligence*. Wuhan, China.

Zhang, C., & Woodland, P. C. (2016). DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions. *IEEE International Conference on Acoustics, Speech and Signal Processing*, (pp.5300–5304). Shanghai, China. IEEE.

Zhu, P., Isaacs, J., Fu, B., & Ferrari, S. (2018). Deep learning feature extraction for target recognition and classification in underwater sonar images, *Conference on Decision and Control*, (pp.2724–2731). Melbourne, VIC, Australia. IEEE.