Construction of CNNs for Abnormal Heart Sound Detection using Data Augmentation

Shumpei Takezaki and Kazuya Kishida

***Abstract*—Cardiovascular diseases (CVDs) are the main cause of deaths all over the world. To detect the abnormalities of a heart automatically, convolutional neural networks (CNNs) learned by using heart sound signals (i.e., the phonocardiogram or PCG) are proposed. Generally, CNNs need sufficient annotated training data to achieve the high performance. However, the annotated PCGs (i.e. PCGs labelled with abnormal or normal) dataset is not sufficient because of personal information and burden of physicians. Therefore, we need to improve the classification performance of CNNs even when annotated PCGs is insufficient. In this paper, in order to solve above problem we consider two data augmentation (DA) methods, one is Window Slicing with Spectrogram (WSS), which slices single PCG to make multiple signals and transforms the signals into spectrogram data, the other is Synthetic Spectrogram based GANs (SSG), which generates synthetic data using generative adversarial networks (GANs). In order to show the validity of considered two DA methods, we perform some experiments concerning heart sounds detection and discuss the results of experiments in point of the accuracy, the sensitivity and the specificity.**

***Index Terms*—Convolutional Neural Networks, Data Augmentation, Generative Adversarial Networks, Heart Sound Detection**

I. Introduction

CARDIOVASCULAR diseases (CVDs) are considered one of major causes of deaths all over the world. The estimated 17.5 million people died due to CVDs, representing 31% of global deaths [1]. Generally, the auscultation by physicians is used to detect heart abnormalities. Thus, physicians need extensive training to develop their expertise in order to understand the auscultation. It is interesting that the diagnosis accuracy of medical students and physicians is between 20-40% and the diagnosis accuracy of expert cardiologists is about 80% [2], [3]. Therefore, we think it is useful to develop the system detecting the abnormalities of a heart accurately.

Some CNNs learned by using heart sound signals (PCG) are proposed to detect heart abnormalities. These methods are that the classifiers constructed by trained CNNs classify PCGs into abnormal or normal. Generally, CNNs need sufficient annotated data to archive the high classification performance. However, annotated PCGs (i.e. PCGs labelled with abnormal or normal) dataset is not sufficient because of personal information and burden of physicians. Therefore, we should need to develop the method constructing the classifier with the high accuracy even when annotated PCGs are insufficient.

A data augmentation (DA) is one of the methods to compensate insufficient training data. DA is able to generate non-real data using real data in order to increase training data for CNNs. DA (e.g. rotation, cutting, e.t.c.) is often leveraged in image recognition using CNNs. Moreover, it is reported that DA does not only improve the classification performance of the classifier using CNNs but also improve the robustness [4].

In this paper, we aim to construct the classifier composed of CNNs classifying PCGs with higher classification performance when the number of PCG is insufficient. We consider two DA methods to compensate insufficient PCGs, (i) Window Slicing with Spectrogram (WSS), which slices single PCG to make multiple signals and transforms the signals into spectrogram data, (ii) Synthetic Spectrogram based GANs (SSG), which generates synthetic data using GANs [5]. We construct the classifier composed of CNNs trained by using training data including training data generated by our considered two DA methods. We evaluate the classification performance of the constructed classifier using multiple evaluation metrics and discuss the validity of the considered two DA methods.

II. Methodology

A. Heart Sound Classification with CNNs

We transform PCGs into the spectrogram data. We construct the classifier composed of CNNs by using the spectrogram data. The constructed classifier classifies spectrogram data into abnormal or normal. Fig.1 shows the process of heart sound classification. We describe the process of constructing the classifier and the process of classifying unannotated PCG (spectrogram data) into abnormal or normal.

[The process of constructing the classifier]

(i) Measure PCGs (wav file) and annotate them abnormal or normal. (We use PCGs provided by the 2016 PhysioNet/Computing Cardiology Challenge [6]. And sampling rate is 2000 Hz.)
(ii) Transform PCGs into spectrogram data. (On the process of transformation, we use first 16,384 samples of a PCG in order to obtain a spectrogram data.)
(iii) Construction of the classifier by training the CNNs using spectrogram data.

[The process of classifying a test PCG]

(i) Measure a test PCG at sampling rate of 2000 Hz. (We use PCGs provided by the 2016 PhysioNet/Computing Cardiology Challenge [6] as unannotated PCG.)

Manuscript received August 6th, 2021; revised August 26th, 2021.

S. Takezaki is with the Advanced Mechanical and Electronic Control Systems Engineering, National Institute of Technology, Kagoshima College, 1460-1 Shinko, Hayato, Kirishima, Kagoshima, 899-5102, Japan, email:shumpeitakezaki@gmail.com

K. Kishida is with the Department of Electronic Control Engineering, National Institute of Technology, Kagoshima College, email:kishida@kagoshima-ct.ac.jp

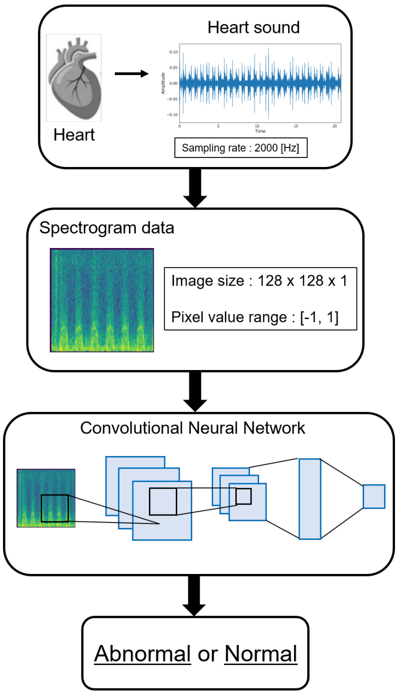


Fig. 1. The Process of Heart Sound Classification

(ii) Transform a test PCG into a spectrogram data. (On the process of transformation, we use first 16384 samples of a test PCG in order to obtain a spectrogram data.)
(iii) The classifier classifies a spectrogram data into abnormal or normal.

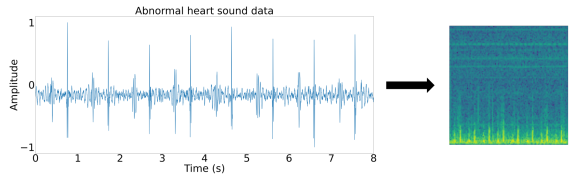
B. Spectrogram Data

Generally, CNNs require two dimensional data like image data as input data. However, a PCG is one dimensional time-series data. So, short time Fourier transformation (STFT) is used to transform one dimensional data into two dimensional data called spectrogram data. CNNs constructed by using spectrogram data of PCGs is leveraged in a heart sound classification [7], [8] and a electrocardiogram classification [9] [10]. STFT is mathematically represented as Equation (1) where $X[m, \omega]$ denotes the spectrogram data (m is x-axis and ω is y-axis), $x[n]$ denotes the original PCG and w denotes the window function. The Window function w is Hamming window given as Equation (2). In Equation (2), the window size M is 256. The x-axis of spectrogram data represents time, y-axis represents frequency, color denotes each frequency density. We use common logarithm against spectrogram data and then normalize spectrogram data to [-1, 1].

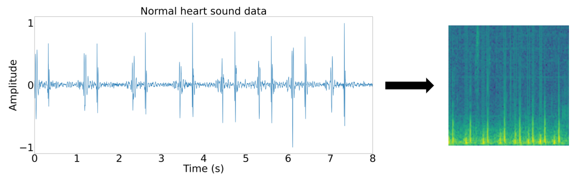
$$X[m, \omega] = \sum_{n=-\infty}^{\infty} |x[n]w(n-m)e^{-j\omega n}| \quad (1)$$

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi k}{M-1}\right), & 0 \le k \le M-1 \\ 0, & \textit{otherwise} \end{cases} \quad (2)$$

Fig.2 shows first 16,384 samples (about first 8 seconds) of PCGs and these spectrogram data. There are abnormal PCG in Fig.2(a) shows a example of an abnormal PCG and an abnormal spectrogram data and Fig.2(b) shows a example of a normal PCG and a normal spectrogram data. The images of spectrogram data in Fig.2(a) and Fig.2(b) are the visualized images from spectrogram data.



(a) Abnormal PCG and Spectrogram Data



(b) Normal PCG and Spectrogram Data

Fig. 2. Examples of PCG and its Spectrogram Data

C. Convolutional Neural Networks

We use CNNs model called ResNet18 [11] (TABLE I) in heart sound classification. The feature of ResNet18 is the skipping connections which are enable to train the deep layer model. In TABLE I, Conv 7×7 represents that the filter size is 7×7 and the stride is 2 in the convolution layer, BN represents the batch normalization [12] and Dense represents the fully connected layer. Fig.3 shows the architecture of ResBlock which has twice of BN, ReLU and convolution layer. And the output y of ResBlock is calculated by adding the output of second convolution layer and the input x of ResBlock.

TABLE I
THE ARCHITECTURE OF RESNET18

Classifier	Activation	Output Shape
Input data	-	$128 \times 128 \times 1$
Conv 7×7	-	$64 \times 64 \times 64$
BN	ReLU	$64 \times 64 \times 64$
Maxpool	-	$32 \times 32 \times 64$
ResBlock	-	$32 \times 32 \times 64$
ResBlock	-	$32 \times 32 \times 64$
ResBlock	-	$16 \times 16 \times 128$
ResBlock	-	$16 \times 16 \times 128$
ResBlock	-	$8 \times 8 \times 256$
ResBlock	-	$8 \times 8 \times 256$
ResBlock	-	$4 \times 4 \times 512$
ResBlock	-	$4 \times 4 \times 512$
GlobalAverage	-	512
Flatten	-	512
Dense	ReLU	512
0.5 Dropout	-	512
Dense	-	1
BN	Sigmoid	1

D. Generative Adversarial Networks

Generative Adversarial Networks (GANs) [5] is the generative model which can generate realistic and diverse images.

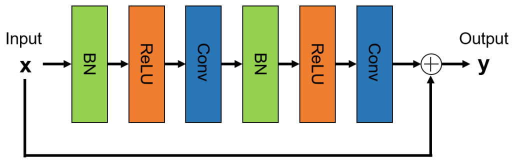


Fig. 3. The Architecture of Resblock

GANs consists of two neural networks, the generator G and the discriminator D in Fig.4. The generator G generates the synthetic data $G(z)$ from noise z. The discriminator D discriminates the inputs x (real data) and $G(z)$ into real data or synthetic data generated by the generator G. The generator G is trained in order to generate the synthetic data that can fool the discriminator D. The discriminator D is trained in order to be able to discriminate real or synthetic data accurately.

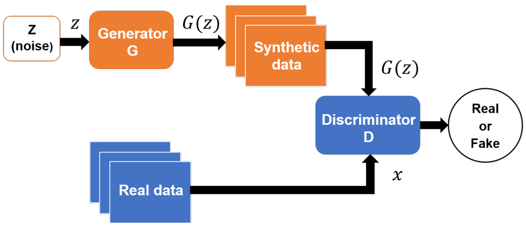


Fig. 4. The Concept Image of GANs

The training of each generator G and discriminator D is described as minimax game shown in Equation (3).

$$\min_G \max_D V(D,G) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log D(\boldsymbol{x})] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_z}[\log(1-D(G(\boldsymbol{z})))] \tag{3}$$

Additionally, each loss function of generator G and discriminator D is described as Equation (4) and Equation (5) respectively.

$$L_G = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_z}[\log(1-D(G(\boldsymbol{z})))] \tag{4}$$

$$L_D = -\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_z}[\log(1-D(G(\boldsymbol{x})))] \tag{5}$$

where $\mathbb{P}_r$ represents the data distribution of real data, $\mathbb{P}_z$ represents the data distribution of noise $z \sim N(0,1)$, $\boldsymbol{x}$ represents the real data, $G(\boldsymbol{z})$ represents the synthetic data generated by the generator G.

We use the generator G (TABLE II) and the discriminator D (TABLE III). In TABLE II and TABLE III, Dense represents the fully connected layer, Conv represents that the filter size is 3×3 and the stride is 2 in the convolution layer, TransConv represents the deconvolution layer that consists of the convolution layer of the filter size 3×3 and the stride 1 and the pooling layer. And BN represents batch normalization [12] that is method in order to stabilize the training of CNNs and SN represents spectral normalization [13] that can stabilize the training of GANs.

III. Data Augmentation

A. Window Slicing with Spectrogram (WSS)

When transforming the PCG into the spectrogram data, a part of data or all data of the PCG are used. Generally, the single PCG is handled to generate the single spectrogram data. So, if we can obtain the multiple spectrogram data from the single PCG, we can increase the number of training data.

Window slicing [14] is proposed as an effective DA corresponding to time-series data like a PCG. Window slicing can make multiple time-series data by slicing single time-series data into a specific length (slice length). So the multiple PCGs are obtained from the single PCG by using window slicing. We call the multiple PCGs made by window slicing, the increased the PCG.

We consider the DA method of transforming the PCGs including the increased the PCG into the spectrogram data receptively. Therefore, we can increase the training data of the spectrogram data. We set the slice length of window slicing to 16384 samples in this paper. In the following, we refer the DA using window slicing as WSS. Further the movement length of each slice is set where slice length $\times$ slice ratio. Slice ratio is the parameter and set through the experiments.

B. Synthetic Spectrogram based GANs (SSG)

As one of the DA methods, the synthetic data generated by GANs is known to be used the training data. GANs based on the DA is often leveraged for compensating insufficient medical data such as the MRI classification [15], the CT classification [16] and so on. CNNs trained by using the synthetic data generated by GANs based on the DA improved the classification performance [4].

We consider the DA method based on GANs which generate the synthetic spectrogram data. We use the synthetic spectrogram data generated by GANs trained using the spectrogram data transformed the original PCG in the training of CNNs.

We need to choose the synthetic data for the training of CNNs because some synthetic data generated by GANs are

TABLE II
The Parameters of Generator

Generator	Activation	Ouput Shape
Latent vector	-	128
Dense	-	16384
Reshape	-	$4 \times 4 \times 1024$
TransConv, BN	ReLU	$8 \times 8 \times 512$
TransConv, BN	ReLU	$16 \times 16 \times 256$
TransConv, BN	ReLU	$32 \times 32 \times 128$
TransConv, BN	ReLU	$64 \times 64 \times 64$
TransConv	Tanh	$128 \times 128 \times 1$

TABLE III
The Parameters of Discriminator

Discriminator	Activation	Ouput Shape
Input data	-	$128 \times 128 \times 1$
Conv, SN	LReLU($\alpha = 0.2$)	$64 \times 64 \times 64$
Conv, SN	LReLU($\alpha = 0.2$)	$32 \times 32 \times 128$
Conv, SN	LReLU($\alpha = 0.2$)	$16 \times 16 \times 256$
Conv, SN	LReLU($\alpha = 0.2$)	$8 \times 8 \times 512$
Conv, SN	LReLU($\alpha = 0.2$)	$4 \times 4 \times 1024$
GlobalsumPool	-	1024
Dense, SN	Sigmoid	1

ISBN: 978-988-14049-1-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

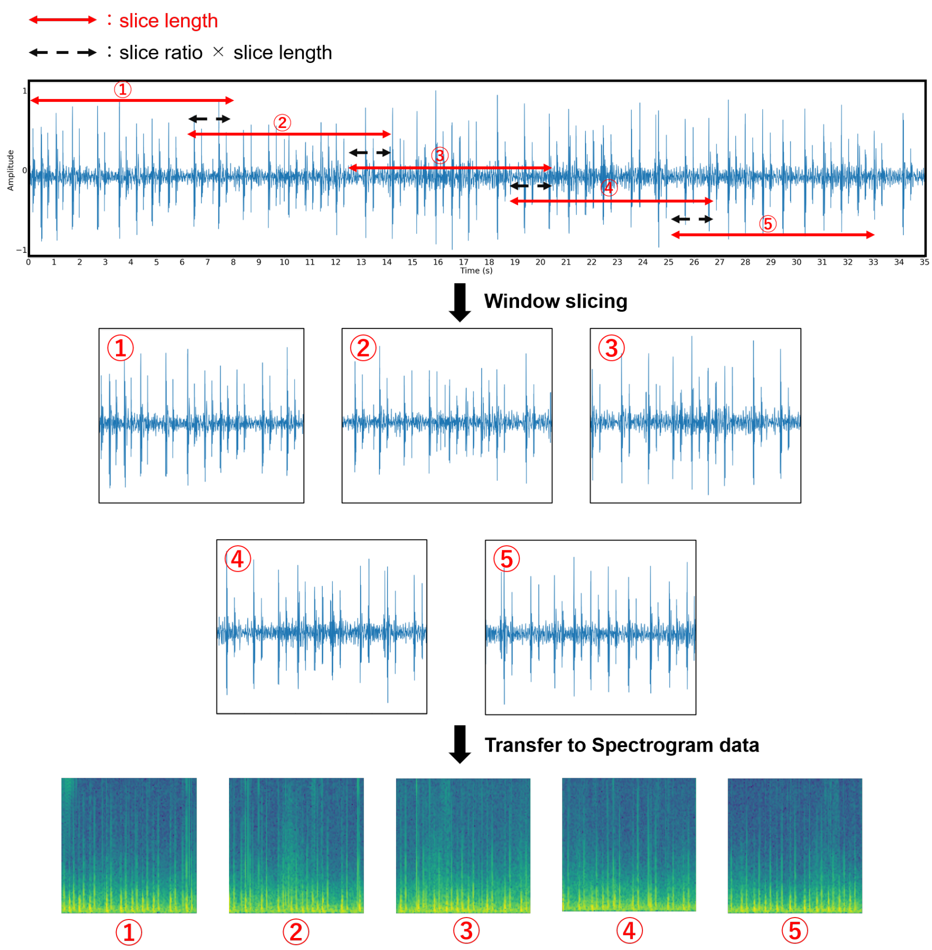


Fig. 5. The Flow of Generating the Spectrogram Data by WSS

obviously different from the original data. Therefore, we generate and choose the synthetic data that is used in the training of CNNs in the following steps.

Step1: The trained GANs generates 100×10^3 synthetic spectrogram data.
Step2: Each synthetic spectrogram data (128×128 dimensional vector) is transformed into the 512 dimensional vector (synthetic vector).
Step3: The original spectrogram data generated by the trained GANs is transformed into the vector (original vector) same as Step2.
Step4: Each synthetic spectrogram data is scored by calculating the *score* using the synthetic vector and the original vectors.
Step5: 5×10^3 (10×10^3) synthetic spectrogram data are choosen from the order of the highest *score*.

In Step1, GANs are trained by using the original spectrogram data in order to generate the synthetic spectrogram data. In Step2, each synthetic spectrogram data generated by the trained GANs is transformed into each vector by using the encorders sucn as CNNs trained using the original spectrogram data. The encorder is composed of the input layer and second layer from the end of fully connected layer of CNNs (The output units of the fully connected layer is 512) and compresses the synthetic spectrogram data into the vector. Therefore, the number of dimensions of each vector equals the outputs of second from the end of fully connected layer and the number of dimensions of the vector is 512. In Step3, each original spectrogram data is transformed into each vector as with Step2. In Step4, *score* of each synthetic spectrogram data is calculated by the sum of the cosine similarity of the synthetic vector and all original vectors as Equation (6).

$$score = \sum_{i}^{N} similarity(\vec{q}, \vec{d_i}) \tag{6}$$

$$similarity(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{7}$$

where $\vec{q}$ represents a synthetic vector, $\vec{d_i}$ represents a original vector and N is the number of the original spectrogram data. And the cosine similarity is derived by Equation (7). We calculated the *score* for 100×10^3 synthetic vectors respectively. Then we choose 5×10^3 (10×10^3) synthetic data from the order of the highest *score* and the chosen synthetic vectors are used as the training data of CNNs. Moreover the synthetic data of abnormal and the synthetic data of normal are separately generated by GANs. Finally, we obtain 5×10^3 (10×10^3) abnormal synthetic spectrogram data and 5×10^3 (10×10^3) normal synthetic spectrogram data.

ISBN: 978-988-14049-1-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Experiment

A. Heart Sound Dataset

We used the PCG dataset provided by the 2016 PhysioNet/Computing in Cardiology Challenge [6]. We made the PCG dataset for experiment (the original dataset) by selecting PCGs that have more than 16384 samples among the PCG data set. And we prepared the original dataset including 558 abnormal PCGs and 2174 normal PCGs.

B. Evaluation method

We compare the classification performance of CNNs constructed by using the considered DA with CNNs constructed not using DA in order to evaluate the effectiveness of the considered DA methods. We use the accuracy, the sensitivity and the specificity as the evaluation metrics. We adopt 5-fold validation.

C. Experiment I: ResNet18

In this Experiment I, we construct the classifier (i.e. CNNs) trained by using only the original training data and evaluate the classification performance of the constructed CNNs. We use CNNs composed of ResNet18 as shown in TABLE I. We use the training data 1968 PCGs (402 abnormal / 1566 normal), the validation data 217 PCGs (44 abnormal / 173 normal) and the test data 347 PCGs (112 normal / 235 normal) by dividing the original PCG dataset in this experiment I. All PCGs are transformed into the spectrogram data. And We use the spectrogram data as the input data of CNNs. The parameters of CNNs are 200 epochs, 128 batch size and 0.001 learning rate in the training of CNNs. And the loss function is binary cross entropy and the optimizer is Adam [17].

We show the result of this experiment I in TABLE IV. From the result of Experiment I in TABLE IV, we can see that the accuracy 91.5 %, the sensitivity 82.2 % and the specificity 93.8 %.

D. Experiment II: WSS

In this Experiment II, we construct the classifiers (i.e. CNNs) trained by using the training data augmented by WSS and the original training data, and evaluate the classification performance of the classifiers. We use the same training data of PCGs, the validation data of PCGs and the test data of PCGs as experiment I. The training data of PCGs are augmented by WSS, that is the augmented training data of PCGs. We use 5 slice ratios of [0, 0.2, 0.4, 0.6, 0.8] as the parameter of WSS. Thus, we generate 5 sets of the augmented training data of PCGs.

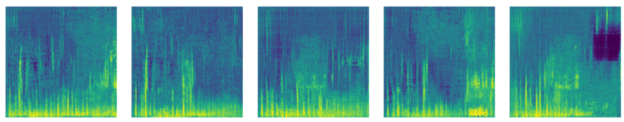
All PCGs are transformed into spectrogram data. And we construct 5 CNNs by using the augmented training data of the spectrogram data and the original training data of spectrogram data. And we evaluate the classification performance of 5 constructed classifiers. We use CNNs composed of ResNet18 as shown in TABLE I. The parameters of each CNNs are 200 epochs, 512 batch size and 0.001 learning rate in the training of CNNs. And the loss function is binary cross entropy and the optimizer is Adam [17].

We show the result of this experiment II in Experiment II of TABLE IV. In TABLE IV, WSS_x represents that WSS uses the slice ratio x. By comparing the result of experiment I with experiment II, the accuracy, the sensitivity and the specificity are improved. Therefore, we think that WSS improves the classification performance of the classifier. And it has a tendency where the sensitivity decreases and the specificity increases when the slice ratio increase. We think it is caused by increasing the difference between the number of the abnormal spectrogram data and of the normal spectrogram data when the slice ratio increases and then CNNs is trained using the normal spectrogram data more times than the abnormal spectrogram data.

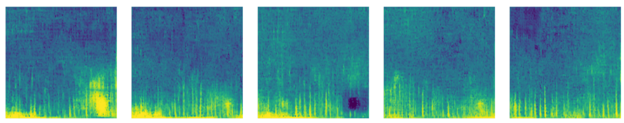
E. Experiment III: SSG

In this Experiment III, we construct the classifiers (i.e. CNNs) trained by using the training data augmented by SSG and the original training data, and evaluate the classification performance of the classifiers. We used the same training data of PCGs, the same validation data PCGs and the same test data PCGs as experiment I. All PCGs are transformed into spectrogram data.

We conduct the training of GANs which is able to generate the synthetic abnormal spectrogram data and the synthetic normal spectrogram data separately. We generate 5×10^3 abnormal and 10×10^3 normal synthetic spectrogram data as the training data after training GANs. Fig.6 shows the examples of the synthetic abnormal spectrogram data (Fig.6-(a)) and the synthetic normal spectrogram data (Fig.6-(b)) respectively. We use CNNs composed of ResNet18 as shown in TABLE I. We use GANs composed of the generator (TABLE II) and the discriminator (TABLE III) and the original training spectrogram data as the training data.The parameter of GANs are 2000 epochs, 32 batch size and 0.001 learning rate in the training of GANs. And the optimizer of GANs is Adam [17]. The parameter of CNNs are 200 epochs, 512 batch size and 0.001 learning rate in the training of CNNs. And the loss function is binary cross entropy and the optimizer is Adam [17].



(a) Abnormal Synthetic Spectrogram Data



(b) Normal Synthetic Spectrogram Data

Fig. 6. Examples of the Visualised Synthetic Spectrogram Data

We show the result of this experiment III in TABLE IV. In TABLE IV, SSG_x represents that the GANs generate the number of x synthetic spectrogram data of abnormal and normal respectively. By comparing the result of experiment I with experiment III, we can see that SSG improves the accuracy, the sensitivity and the specificity. While, we compare SSG_5000 with SSG_10000 and the accuracy was not

ISBN: 978-988-14049-1-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE IV
RESULTS OF EXPERIMENTS

DA set up	Accuracy (%)	Sensitivity (%)	Specificity (%)
Experiment I			
Not using	91.5	82.2	93.8
Experiment II			
WSS_0	93.1	90.8	93.7
WSS_0.2	92.9	87.4	94.3
WSS_0.4	91.7	88.7	92.4
WSS_0.6	93.2	85.4	95.1
WSS_0.8	93.7	84.8	95.9
Experiment III			
SSG_5000	92.6	84.2	94.8
SSG_10000	92.5	86.3	94.0

improved despite of increasing the number of the synthetic data as using training data of CNNs. Additionally, the accuracy of SSG decreases by about 1 % compared with the accuracy of WSS.

V. CONCLUSION

In this study, we considered two DA methods, WSS and SSG, in order to improve the classification performance of the classifier (CNNs) for heart sound classification when the PCGs are insufficient. We constructed the classifiers by using the training data augmented by WSS and SSG respectively. From the results of some experiments, we knew that the accuracy, the sensitivity and the specificity of the constructed classifiers are improved. Therefore, we confirmed the validity of our considered DA methods for improving the performance of heart sound classification.

In our future works, we consider the method combined WSS and SSG, the improvement of the architecture and the training methods of GANs, the selection method of synthetic data, the construction of the classifier corresponding the variable length data.

REFERENCES

[1] "Who. 2016 world statistics on cardiovascular disease," who.int/mediacentre/factsheets/fs317/en/.
[2] S. K. Gardezi, S. G. Myerson, J. Chambers, S. Coffey, J. d'Arcy, F. R. Hobbs, J. Holt, A. Kennedy, M. Loudon, A. Prendergast *et al.*, "Cardiac auscultation poorly predicts the presence of valvular heart disease in asymptomatic primary care patients," *Heart*, vol. 104, no. 22, pp. 1832–1835, 2018.
[3] F. Demir, A. Şengür, V. Bajaj, and K. Polat, "Towards the classification of heart sounds based on convolutional deep neural network," *Health information science and systems*, vol. 7, no. 1, pp. 1–9, 2019.
[4] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
[6] C. Liu, D. Springer, Q. Li, B. Moody, R. A. Juan, F. J. Chorro, F. Castells, J. M. Roig, I. Silva, A. E. Johnson *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiological Measurement*, vol. 37, no. 12, p. 2181, 2016.
[7] T. Nilanon, J. Yao, J. Hao, S. Purushotham, and Y. Liu, "Normal/abnormal heart sound recordings classification using convolutional neural network," in *2016 computing in cardiology conference (CinC)*. IEEE, 2016, pp. 585–588.
[8] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, and K. Sricharan, "Recognizing abnormal heart sounds using deep learning," *arXiv preprint arXiv:1707.04642*, 2017.
[9] K. N. Khan, F. A. Khan, A. Abid, T. Olmez, Z. Dokur, A. Khandakar, M. E. Chowdhury, and M. S. Khan, "Deep learning based classification of unsegmented phonocardiogram spectrograms leveraging transfer learning," *arXiv preprint arXiv:2012.08406*, 2020.
[10] J. Huang, B. Chen, B. Yao, and W. He, "Ecg arrhythmia classification using stft-based spectrogram and convolutional neural network," *IEEE Access*, vol. 7, pp. 92 871–92 880, 2019.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
[13] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.
[14] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *arXiv preprint arXiv:1603.06995*, 2016.
[15] C. Han, L. Rundo, R. Araki, Y. Nagano, Y. Furukawa, G. Mauri, H. Nakayama, and H. Hayashi, "Combining noise-to-image and image-to-image gans: brain mr image augmentation for tumor detection," *IEEE Access*, vol. 7, pp. 156 966–156 977, 2019.
[16] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.
[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.