Data Augmentation and the Improvement of the Performance of Convolutional Neural Networks for Heart Sound Classification

Shumpei Takezaki and Kazuya Kishida

Abstract—The effective management and early diagnosis of cardiovascular diseases (CVDs) are crucial to bring down the mortality associated with these diseases. Because detecting CVDs can be a difficult task, especially when no symptoms are present, developing systems that can detect CVDs or heart abnormalities automatically has attracted significant research attention in recent times. For this purpose, several convolutional neural networks (CNNs) learned using heart sound signals (i.e., the phonocardiogram or PCG) have been proposed. Generally, CNNs require a high volume of annotated training data to achieve high performance. However, annotated PCG (i.e., PCGs labelled as abnormal or normal) dataset is not sufficient for training CNNs. To address this issue, the classification performance of CNNs need to be improved, so that they can be trained even on an insufficient PCG database. In this study, we consider two data augmentation (DA) methods: window slicing with spectrogram, which slices a single PCG to generate multiple signals and transforms the signals into spectrogram data; the other is a synthetic spectrogram based generative adversarial network, which generates synthetic data. To demonstrate the validity of the two DA methods, we performed experiments concerning heart sound detection and discussed the results of the experiments in terms of accuracy, sensitivity, and specificity.

Index Terms—convolutional neural networks, data augmentation, generative adversarial networks, heart sound detection

I. INTRODUCTION

Cardiovascular diseases (CVDs) represent a major cause of mortality worldwide. In 2019 alone, an estimated 17.9 million people died of CVDs, representing 32% of the global deaths [1]. In general, physicians detect heart abnormalities using auscultation. Understanding and interpreting auscultation require a high degree of expertise: This can be gleaned from the fact that the diagnostic accuracy of medical students and physicians based on auscultation is between 20-40%, and that of expert cardiologists is approximately 80% [2], [3]. Therefore, we believe that it is very useful to develop a system that detects heart abnormalities more accurately than expert cardiologists.

For this purpose, several convolutional neural networks (CNNs) that learn using heart sound signals (PCG) have been proposed [3]–[5]. These methods construct classifiers with trained CNNs and classify PCGs as abnormal or normal. Generally, CNNs require high volumes of annotated data to achieve high classification performance. However, annotated PCGs (i.e., PCGs labelled with abnormal or normal) datasets are not always sufficient because the datasets predominantly contain patients’ personal information, which cannot be used without consent, and burden of physicians. Therefore, it is necessary to develop a method for constructing a classifier with high accuracy, even when trained on the insufficient annotated PCG dataset [6].

Data augmentation (DA) is a method used to compensate for insufficient training data. DA generates additional data and completely new data using real data to increase the training data for the CNNs, and in turn, improve the performance of CNNs. DA (e.g., rotation and cutting) is often leveraged in CNN-based image recognition. Moreover, it has been reported that DA not only improves the classification performance of the CNNs, but also improves their robustness [7].

In this study, we aim to construct a CNNsaa classifier to classify PCGs with higher performance when the number of PCGs is insufficient. We consider two DA methods to compensate for insufficient PCGs: (i) window slicing with a spectrogram (WSS), which slices a single PCG to generate multiple signals and transforms the signals into spectrogram data, and (ii) synthetic spectrogram based generative adversarial network (SSG), which generates synthetic data using GANs [8]. We constructed a CNN classifier trained on data generated by our two DA methods. We then evaluated the classification performance of the constructed classifier using multiple evaluation metrics and discussed the validity of the two DA methods.

II. METHODOLOGY

A. Heart Sound Classification with CNNs

Fig.1 shows the heart sound classification process. The PCGs were transformed into spectrogram data. We constructed a CNN classifier using spectrogram data. The constructed classifier classifies spectrogram data as abnormal or normal. We describe the process of constructing the classifier and the process of classifying unannotated spectrogram data as abnormal or normal.

[ 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 [9]. The sampling rate was set to 2000 Hz.)
(ii) Transform PCGs into spectrogram data. (For the process of transformation, we use first 16,384 samples of a PCG to obtain a spectrogram data.)

(iii) Construction of the classifier by training the CNNs using spectrogram data.

[ 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 [9] as unannotated PCG.)
(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, such as image data, as input. However, a PCG is a one-dimensional time-series data. Therefore, the short-time Fourier transform (STFT) is used to transform one-dimensional data into two-dimensional data, called spectrogram data. CNNs constructed using spectrogram data of PCGs are leveraged in heart sound classification [4], [10] and electrocardiogram classification [5], [11]. The STFT is mathematically represented as Equation (1), where \( X[m, \omega] \) denotes the spectrogram data (\( m \) is the x-axis and \( \omega \) is the y-axis), \( x[n] \) denotes the original PCG, and \( w \) denotes the window function. The window function \( w \) is the Hamming window given by Equation (2). In Equation (2), window size \( M \) is 256. The x-axis of the spectrogram data represents time, the y-axis represents frequency, and color denotes each frequency density. We use a common logarithm against the spectrogram data and then normalize the spectrogram data to [-1, 1].

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

\[
w(n) = \begin{cases} 
0.54 - 0.46 \cos \left( \frac{2\pi k}{M - 1} \right), & 0 \leq k \leq M - 1 \\
0, & \text{otherwise}
\end{cases}
\]

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

C. Convolutional Neural Networks

We used a CNNs model called ResNet18 [12] (TABLE I) for heart sound classification. The feature of ResNet18 is the skipping connections that can train the deep layer model. In TABLE I, Conv \( 7 \times 7 \) indicates that the filter size is \( 7 \times 7 \) and the stride is 2 in the convolution layer, BN represents batch normalization [13] and dense represents the fully connected layer. Fig.3 shows the architecture of ResBlock which has twice of BN, ReLU and convolution layer [14]. The output y of ResBlock is calculated by adding the output of the second convolution layer to the input x of ResBlock.

D. Generative Adversarial Networks

GANs [8] are generative models that can generate realistic and diverse images. GANs consist of two neural networks: generator \( G \) and discriminator \( D \), as shown in Fig.4. Generator \( G \) generates synthetic data \( G(z) \) from noise
Fig. 3. The Architecture of Resblock

z. Discriminator $D$ discriminates the inputs $x$ (real data) and $G(z)$ into the real or synthetic data generated by generator $G$ into fake. Generator $G$ is trained to generate synthetic data that can fool discriminator $D$. Discriminator $D$ is then trained to accurately discriminate between real and synthetic data.

The training for both generator $G$ and discriminator $D$ is described by the minimax game shown in Equation (3).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_x} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))]$$

(3)

Additionally, each loss function of the generator $G$ and discriminator $D$ is described by Equations (4) and (5), respectively.

$$L_G = \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))]$$

(4)

$$L_D = -\mathbb{E}_{x \sim P_x} [\log D(x)] - \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))]$$

(5)

where $P_x$ represents the data distribution of the real data, $P_z$ represents the data distribution of the noise $z \sim N(0, 1)$,

represents the real data, $G(z)$ represents the synthetic data generated by the generator $G$.

We used generator $G$ (TABLE II) and discriminator $D$ (TABLE III). In TABLE II and TABLE III, Dense represents the fully connected layer, Conv represents the convolution layer that has the filter size $3 \times 3$ and the stride 2, and TransConv represents the deconvolution layer that consists of the convolution layer of the filter size $3 \times 3$ and stride 1 and the pooling layer. BN represents batch normalization [13], which is a method for stabilizing the training of CNNs, and SN represents spectral normalization [15], which can stabilize the training of GANs.

### III. DATA AUGMENTATION

#### A. Window Slicing with Spectrogram (WSS)

When transforming the PCG into spectrogram data, a part or the entire data of the PCG are used. In general, a single PCG is used to generate a single spectrogram data point. Therefore, if we can obtain multiple spectrogram data points from a single PCG, we can increase the amount of training data.

Window slicing [16] has been proposed as an effective DA technique for time-series data such as a PCG. Window slicing generates multiple time-series data by slicing single time-series data into a specific length (slice length). Therefore, multiple PCGs were obtained from a single PCG using window slicing. We refer to the multiple PCGs made by window slicing as increased PCG.

We consider the DA method of transforming the PCGs, including the increased PCG, into spectrogram data, as shown in Fig. 5. Therefore, we can increase the training data of the spectrogram data. We set the length of the window slicing to 16384 samples in this study. In the following, we refer to DA using window slicing as the WSS. Further, the
movement length of each slice was set as the slice length $\times$ slice ratio. The slice ratio is the parameter and is set through the experiments.

### B. Synthetic Spectrogram based GANs (SSG)

Synthetic data generated by GANs have been used as training data in many studies. Furthermore, DA through GANs is often leveraged for compensating insufficient medical data in tasks such as MRI classification [17] and CT classification [18]. In several studies, CNNs trained using synthetic data generated by GANs have shown improved classification performance [7].

In this study, we consider a DA method based on GANs that generates synthetic spectrogram data, which we subsequently use to transform the original PCG dataset into a dataset sufficient for training CNNs.

It is important to note, however, it is critical to choose specific synthetic data for training CNNs because some synthetic data generated by GANs are obviously different from the original data. Therefore, we generate and choose the synthetic data to be used for training the CNNs in the following steps.

1. The trained GANs generate synthetic spectrogram data.
2. The synthetic spectrogram data (128 $\times$ 128 dimensional vector) are transformed into a 512-dimensional vector (synthetic vector).
3. All original spectrogram data are transformed into the vectors (original vector) in the same way as in Step 2.
4. The synthetic spectrogram data are scored by calculating the score using the synthetic as well as original vectors.
5. Step 1 $\sim$ Step 4 are repeated $10^5$ times.
6. $5 \times 10^3$ ($10 \times 10^3$) synthetic spectrogram data are chosen from the order of the highest score.

In Step 1, the trained GANs generate the synthetic spectrogram data. In Step 2, the synthetic spectrogram data are transformed into a vector (synthetic vector) using the encoder, which is a CNN trained using the original spectrogram data. The encoder is a trained Resnet18, as shown in TABLE I; the
A. Heart Sound Dataset

We used the PCG dataset provided by the 2016 PhysioNet/Computing in Cardiology Challenge [9]. The dataset consisted of six subdatasets, which were collected in different environments and using different devices.

We created a PCG dataset for the experiments (original dataset). We created the original dataset by selecting PCGs that had more than 16384 samples among the PCG dataset. We prepared the original dataset, which included 558 abnormal PCGs and 2174 normal PCGs.

B. Evaluation Method

We evaluated the efficiency of WSS and SSG in terms of classification performance. We compared the classification performance of CNNs constructed using DA with CNNs constructed without using DA to evaluate the effectiveness of the proposed DA methods.

Accuracy, sensitivity, and specificity were used as evaluation metrics. We adopted 5-fold cross validation to evaluate the classification performance. For a better evaluation, we adopted window slicing (slice ratio of 0. i.e., WSS_0) to test PCGs. Therefore, the size of the test spectrogram data increases, which leads to better evaluation.

We performed McNemar’s test on the classification results with our considered DA and without one to confirm statistically significant improvement. In McNemar’s test, we used the collection of classification results for each fold of test data.

The data distributions were visualized using t-SNE [19]. The purpose is to compare the data distributions of the original data and synthetic data of the considered DAs. t-SNE is based on visualization methods and reduces the high dimensionality into a low one (two or three dimensionality). We randomly selected 300 spectrogram data points from each of the original data, WSS synthetic data, and SSG synthetic data, and adapted t-SNE on these spectrogram data.

C. Experiment I: ResNet18

In Experiment I, we constructed a classifier (i.e., CNNs) trained using only the original training data and evaluated the classification performance of the constructed CNNs. We used CNNs composed of ResNet18, as shown in TABLE I. Furthermore, in each fold, we used the training data of 1968 PCGs (402 abnormal / 1566 normal), validation data of 217 PCGs (44 abnormal / 173 normal), and test data of 547 PCGs (112 normal / 435 normal) by dividing the original PCG dataset in experiment I. All PCGs were transformed into spectrogram data. We used spectrogram data as the input data of the CNNs. The parameters of the CNNs were 200 epochs, 128 batch sizes, and 0.001 learning rates in the training of CNNs. The loss function is the binary cross-entropy, and the optimizer is Adam [20].

The results of Experiment I are shown in TABLE IV. From the results of Experiment I in TABLE IV, we can see that the accuracy, sensitivity, and specificity were 91.5 %, 82.2 %, and 93.8 %, respectively.

D. Experiment II: WSS

In Experiment II, we constructed classifiers (i.e., CNNs) trained using the training data augmented by WSS and

### Table IV: Results of Classification Performances on Experiments

<table>
<thead>
<tr>
<th>DA set up</th>
<th>Accuracy (%)</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not using</td>
<td>91.5</td>
<td>82.2</td>
<td>93.8</td>
<td>-</td>
</tr>
<tr>
<td>Experiment II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSS_0</td>
<td>93.1</td>
<td>90.8</td>
<td>93.7</td>
<td>0.367</td>
</tr>
<tr>
<td>WSS_0.2</td>
<td>92.9</td>
<td>87.4</td>
<td>94.3</td>
<td>7.25 × 10^{-3}</td>
</tr>
<tr>
<td>WSS_0.4</td>
<td>91.7</td>
<td>88.7</td>
<td>92.4</td>
<td>0.876</td>
</tr>
<tr>
<td>WSS_0.6</td>
<td>93.2</td>
<td>85.4</td>
<td>95.1</td>
<td>1.97 × 10^{-6}</td>
</tr>
<tr>
<td>WSS_0.8</td>
<td>93.7</td>
<td>84.8</td>
<td>95.9</td>
<td>1.27 × 10^{-7}</td>
</tr>
<tr>
<td>Experiment III</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSG_5000</td>
<td>92.6</td>
<td>84.2</td>
<td>94.8</td>
<td>1.14 × 10^{-5}</td>
</tr>
<tr>
<td>SSG_10000</td>
<td>92.5</td>
<td>86.3</td>
<td>94.0</td>
<td>6.08 × 10^{-3}</td>
</tr>
</tbody>
</table>

output of the 512-dimensional vector is used as a synthetic vector, which is the output of the second layer from the end of the fully connected layer (the output units of the fully connected layer are 512). In Step3, all the original spectrogram data are transformed into vectors (original vectors) as in Step2. In Step4, score of the synthetic spectrogram data is calculated, as shown in Fig. 6. This calculation is the sum of the cosine similarity of the synthetic vector and all the original vectors, as in Equation (6).

\[
\text{score} = \sum_{i} \text{similarity}(\vec{q}, \vec{d}_i) \tag{6}
\]

where \(\vec{q}\) represents the synthetic vector, \(\vec{d}_i\) represents the original vector, and N is the amount of original spectrogram data. The cosine similarity was derived using Equation (7).

\[
\text{similarity}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \tag{7}
\]
TABLE V
THE NUMBER OF AUGMENTED DATA OF WSS

<table>
<thead>
<tr>
<th>WSS setup</th>
<th>Fold1 Abnormal</th>
<th>Normal</th>
<th>Fold2 Abnormal</th>
<th>Normal</th>
<th>Fold3 Abnormal</th>
<th>Normal</th>
<th>Fold4 Abnormal</th>
<th>Normal</th>
<th>Fold5 Abnormal</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSS_0</td>
<td>833</td>
<td>2319</td>
<td>832</td>
<td>2297</td>
<td>828</td>
<td>2325</td>
<td>812</td>
<td>2278</td>
<td>814</td>
<td>2276</td>
</tr>
<tr>
<td>WSS_0.2</td>
<td>1132</td>
<td>3085</td>
<td>1132</td>
<td>3069</td>
<td>1120</td>
<td>3559</td>
<td>1106</td>
<td>3047</td>
<td>1110</td>
<td>3036</td>
</tr>
<tr>
<td>WSS_0.4</td>
<td>1500</td>
<td>4333</td>
<td>1500</td>
<td>4322</td>
<td>1491</td>
<td>4364</td>
<td>1472</td>
<td>4293</td>
<td>1464</td>
<td>4274</td>
</tr>
<tr>
<td>WSS_0.6</td>
<td>2376</td>
<td>6906</td>
<td>2374</td>
<td>6870</td>
<td>2358</td>
<td>6953</td>
<td>2326</td>
<td>6839</td>
<td>2324</td>
<td>6812</td>
</tr>
<tr>
<td>WSS_0.8</td>
<td>4918</td>
<td>14582</td>
<td>4928</td>
<td>14512</td>
<td>4897</td>
<td>14667</td>
<td>4839</td>
<td>14444</td>
<td>4820</td>
<td>14389</td>
</tr>
</tbody>
</table>

Fig. 7. Data Distribution of original and WSS by t-SNE

We show the results of Experiment II in TABLE IV. In TABLE IV, WSS_x indicates that WSS uses the slice ratio x. By comparing the results of Experiment I with those of Experiment II, the accuracy, sensitivity, and specificity are improved. We believe that WSS improves the classification performance of the classifier. WSS shows statistical significance, except for WSS_0 and WSS_0.4, and boldface indicates statistical significance in TABLE IV (threshold p-value < 0.05).

The classification performance has a tendency where the sensitivity decreases and specificity increases when the slice ratio increases. We believe that this is caused by increasing the difference between the number of abnormal spectrogram data and normal spectrogram data when the slice ratio increases; then, CNNs are trained using the normal spectrogram data more times than the abnormal spectrogram data.

Fig. 7 shows that the data distribution of original data and augmented data of WSS. There are the corresponding of the points and data in Fig. 7. Abnormal and normal data largely overlap with the original data. Augmented WSS data compensate for the original abnormal and normal data. Therefore, we think that WSS is an augmentation function interpolation of the original data, and CNNs can train using training data that have a dense distribution. This leads to an improvement in heart sound classification performance.

E. Experiment III: SSG

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

We conduct the training of GANs that can generate synthetic abnormal spectrogram data and synthetic normal spectrogram data separately. We generated $5 \times 10^3$ abnormal and $10 \times 10^3$ normal synthetic spectrogram data as the training data after training the GANs. Fig.8 shows the examples of the synthetic abnormal spectrogram data (Fig.8-(a)) and the synthetic normal spectrogram data (Fig.8-(b)). We used CNNs composed of ResNet18, as shown in TABLE I. We used GANs composed of the generator (TABLE II) and discriminator (TABLE III) and the original training spectrogram data as the training data, and the parameters used for training the GANs were: 2000 epochs, 32 batch size, and 0.001 learning rate. Adam was used as the optimizer [20]. The parameters for training the CNNs were: 200 epochs, 512 batch size, and 0.001 learning rate. The loss function is the binary cross-entropy, and the optimizer is Adam [20].

We show the results of Experiment III in TABLE IV. In TABLE IV, SSG_x indicates that the GANs generate the number of x synthetic spectrogram data of abnormal and normal, respectively. By comparing the results of Ex-
V. CONCLUSION

To improve the performance of CNNs for heart sound classification even when the PCGs are insufficient, we considered two DA methods, WSS and SSG. The two DA methods succeeded in augmenting the training data; we then constructed CNN classifiers using the training data augmented by WSS and SSG. As per our experimental results, the classifiers showed improved accuracy, sensitivity, and specificity. Therefore, we confirmed the validity of the two DA methods for improving heart sound classification performance.

In future, we intend to develop a method that combines WSS and SSG because we believe that WSS and SSG complement each other. Moreover, we plan to extend the proposed methods for certain types of bio-signals, such as electrocardiograms, electromyograms, and electroencephalographs.

REFERENCES


