*electronics*

MDPI

*Article*

# Enhanced Speech Emotion Recognition Using DCGAN-Based Data Augmentation

**Ji-Young Baek [1] and Seok-Pil Lee [2,*]**

[1] Department of Computer Science, Graduate School, Sangmyung University, Seoul 03016, Republic of Korea; b00217@naver.com
[2] Department of Intelligent IoT, Sangmyung University, Seoul 03016, Republic of Korea
* Correspondence: esprit@smu.ac.kr

**Abstract:** Although emotional speech recognition has received increasing emphasis in research and applications, it remains challenging due to the diversity and complexity of emotions and limited datasets. To address these limitations, we propose a novel approach utilizing DCGAN to augment data from the RAVDESS and EmoDB databases. Then, we assess the efficacy of emotion recognition using mel-spectrogram data by utilizing a model that combines CNN and BiLSTM. The preliminary experimental results reveal that the suggested technique contributes to enhancing the emotional speech identification performance. The results of this study provide directions for further development in the field of emotional speech recognition and the potential for practical applications.

**Keywords:** artificial intelligence; deep learning; DCGAN; data augmentation; speech emotion recognition

check for updates

**Citation:** Baek, J.-Y.; Lee, S.-P. Enhanced Speech Emotion Recognition Using DCGAN-Based Data Augmentation. *Electronics* **2023**, *12*, 3966. https://doi.org/10.3390/electronics12183966

Academic Editor: George A. Tsihrintzis

Received: 31 August 2023
Revised: 18 September 2023
Accepted: 19 September 2023
Published: 20 September 2023

## 1. Introduction

Speech recognition technology plays a crucial role in enriching and facilitating more intuitive human–machine interactions. Notably, the significance of emotional speech recognition is evident from its broadening applications across various domains including smart homes, healthcare, entertainment, customer service, and sentiment analysis. Initially, emotion recognition in speech research focused on probabilistic models like hidden Markov models (HMMs) and Gaussian mixture models (GMMs) [1–5]. With the emergence of deep learning, the study of emotion recognition through neural networks has become prevalent [6–11]. Nevertheless, due to the complexity and diversity of emotions and the challenge of subjective evaluation, precise emotional speech recognition remains a major challenge.

One of the major challenges hindering progress in these studies is the limited availability of high-quality emotional speech data. In the realm of image processing, popular datasets such as CIFAR10 [12], ImageNet [13], and MNIST [14] have been extensively utilized to train deep learning models. However, these large-scale datasets are inadequate for emotional speech datasets. Among emotional speech datasets, datasets such as IEmoCAP [15], EmoDB [16], and RAVDESS [17] are frequently used in research, but they are relatively small compared with image datasets. To overcome limitations in data, we propose utilizing deep convolutional generative adversarial networks (DCGANs) [18] to augment speech data in the form of mel-spectrograms. While primarily used for image data augmentation, this work explores the application of DCGANs to mel-spectrograms, which are time-frequency representations of speech effectively capturing different components of emotion.

In addition, this study investigates the effectiveness of using a combination of convolutional neural networks (CNNs) and bidirectional long short-term memory (BiLSTM) [19] to accurately identify emotions from mel-spectrogram data. The combination of these techniques indicates the future direction of emotional speech recognition technology development and real-world applications. It provides significant potential for improving the performance of emotional speech recognition.

The structure of this paper is as follows. Initially, prior research on GANs, DCGANs, and mel-spectrograms is examined. The following section outlines the methodology by providing details on the utilized database, data preprocessing, data augmentation through DCGAN, and model design via CNN+BiLSTM. Then, in the Experiments and Results section, we validate the performance of the proposed approach through diverse experiments. In the Discussion section, we analyze the implications of these outcomes, limitations of the research, and potential avenues for future studies.

## 2. Related Work

### *2.1. GAN and DCGAN*

Generative adversarial networks (GANs) were first introduced by Goodfellow et al. in [20]. The central idea behind GANs involves using two neural networks, a generator $G$, and a discriminator $D$, in a competitive game. The generator $G$ strives to create data using noise $z$ from the latent space. Its aim is to produce data that best represent the given noise. Meanwhile, the discriminator attempts to determine if the input data are genuine or generated by the generator. The training of a GAN follows the minimax game format and strives to optimize the objective function presented in Equation (1).

$$\underset{G}{min}\,\underset{D}{max}\, V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

where $E$ denotes the anticipated value and $p_{\text{data}}(x)$ refers to the distribution of the real data, while $p_z(z)$ represents the distribution of the input noise. During the training, the generator strives to produce superior data to deceive the discriminator, whereas the discriminator endeavors to better differentiate the generator output. However, during the learning phase, typical GANs encounter mode collapse, unstable learning dynamics, and difficulty generating high-resolution images.

DCGAN is a variation of GAN proposed by Radford et al. in [18]. This structure efficiently learns high-dimensional image features by incorporating convolutional networks in both the generator and discriminator. The pooling layer is substituted by stride convolutions, which are executed as fractional-strided convolutions or deconvolutions [21] in the generator. In addition, each layer of the network undergoes batch regularization [22] to enhance learning stability. The generator layer implements Rectified linear unit (ReLU) activation functions [23], while the discriminator uses Leaky ReLU activation functions [24]. To boost image generation quality, fully connected layers are minimized or eliminated, and dropouts [25] are not utilized in the DCGAN design. These characteristics significantly contribute to DCGAN's ability to produce high-quality images with high resolution.

In this study, we utilize DCGAN for emotional speech recognition research to produce mel-spectrogram data and investigate how the model's performance can be enhanced by utilizing the generated data. DCGAN has been successfully applied to image data, and our goal is to apply it to speech data for better performance in emotional speech recognition. Combining the generated data with existing data will improve the generalization of the models during training.

### *2.2. Speech Feature Extraction Using Mel-Spectrograms*

Speech data include diverse patterns and changing information over time. One effective way to capture these patterns is by using a mel-spectrogram. A mel-spectrogram shows a visual representation of the changes in frequency over time. The distinction from a conventional spectrogram lies in the conversion of frequency employing a mel scale. This mel scale detects more intricate details at lower frequencies and simpler details at higher frequencies, similar to the characteristics of human hearing.

The first step in calculating the mel-spectrogram is to perform a short-time Fourier transform (STFT) on the audio signal to obtain a time-frequency spectrogram. Mathematically, this is represented by Equation (2):

$$S(f,t) = |STFT(x(t))|^2 \tag{2}$$

Next, the calculated spectrogram undergoes a mel filter bank process to extract the energy in each mel frequency region. The mel filter bank consists of peak-shaped filters that respond to specific mel frequency ranges. The equation for converting the frequencies to the mel scale is defined by Equation (3).

$$Mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \tag{3}$$

One can use this formula to calculate the center frequency of each mel filter. Afterward, one can associate each frequency domain in the spectrogram with the corresponding mel scale and aggregate the energies to produce a mel-spectrogram.

In this study, CNNs are utilized to extract spatial features from speech via a mel-spectrogram. CNNs, capable of learning local features of 2D data like images or spectrograms through multiple layers, are effective in this context. Furthermore, BiLSTM is employed to capture sequential patterns, particularly in speech time series data. BiLSTM, a bidirectional version of long short-term memory (LSTM) [26], learns sequential patterns by taking into account both the preceding and subsequent data points. This enables the model to precisely detect nuanced shifts in emotions or varied speech patterns. Therefore, the model in this study uses a combination of CNNs and BiLSTM to effectively detect and analyze complex patterns and emotional changes in speech data.

### *2.3. Convolutional Neural Networks*

Convolutional neural networks (CNNs) are deep neural network architectures that primarily extract spatial features from 2D data and work well with data forms such as images or spectrograms. CNNs consist of multiple layers, each of which is used to detect and extract patterns in surrounding pixels in the input data. These CNNs are widely used in a variety of fields, including computer vision, natural language processing, and speech processing, and are particularly good at extracting features from 2D data. CNNs consist of the following main layers: convolutional layers, pooling layers, and fully connected layers.

The first layer, the convolutional layer, plays a crucial role in detecting local patterns or features within the input data. In CNNs, multiple filters are applied to the input data, with each filter scanning the data to detect specific patterns. In image processing, these patterns are primarily associated with edges, textures, or meaningful high-level features. For audio data, they are employed to capture variations over time and frequency. Next, the pooling layers are used to reduce the size of feature maps and decrease the computational load. In these layers, operations such as max pooling and average pooling are primarily employed to subsample and abstract features. Max pooling involves selecting the maximum value within a local region of a feature map, thereby retaining the most important features within that region. This process effectively reduces spatial dimensions while enhancing network efficiency by preserving critical information and reducing computational burden. Conversely, average pooling calculates the average value within a local region, resulting in smoother and more generalized feature representations. This abstraction of features enhances the network's robustness to variations in input data. Lastly, the fully connected layers, often structured as a multi-layer perceptron (MLP), utilize the features extracted from the previous layers to perform prediction or classification tasks. These layers serve as connectors between the hierarchically learned features from earlier layers and the final decision-making process. Each neuron within the fully connected layers is connected to every neuron in the preceding layer, allowing for comprehensive interactions and information integration. Through weighted connections and activation functions, these layers transform high-level abstract representations of input data into meaningful predictions and classifications. In summary, fully connected layers aggregate the knowledge acquired by the network during training and capture complex relationships and patterns within the data. Consequently, CNNs excel at making meaningful predictions based on learned features and are particularly effective in tasks such as image recognition, where understanding intricate visual patterns and assigning appropriate labels is crucial.

### *2.4. Bidirectional Long Short-Term Memory*

BiLSTM is an extension of the LSTM model that is specifically designed to capture dependencies and context in sequential data. The model has the ability to process information in both forward and backward directions, allowing for a more complete understanding of the sequence. This makes it particularly useful for tasks such as speech recognition and language translation. Unlike traditional LSTM, which processes data unidirectionally, BiLSTM operates bidirectionally and considers both past and future information at every time step. The basic structure of BiLSTM comprises two LSTM networks. This bidirectional approach enables BiLSTM to integrate context from both directions and create a more thorough comprehension of the input data. One processes the sequence forward, while the other processes it backward. At each time step, the forward LSTM cell processes the sequence from the beginning, and the backward LSTM cell processes it in reverse. The results from both cells are merged via concatenation to generate a conclusive representation of data at the given time interval. This merged representation encompasses details on how every element in the sequence is linked to its past and future context, making BiLSTM highly practical for tasks that entail capturing intricate dependencies. One of the main advantages of BiLSTM is its ability to effectively model and capture long-range dependencies, making it a suitable choice for various types of sequential data analysis tasks, including natural language processing (NLP) applications. The bidirectional nature of BiLSTM enables it to perform exceedingly well in situations where context comprehension from both directions is critical, such as in speech recognition, time series forecasting, and more.

In brief, the BiLSTM model is a flexible deep learning algorithm created to amplify the depiction and interpretation of sequential data by taking into account the past and future context, thus rendering it useful for various applications beyond NLP.

### *2.5. Emotional Speech Database*

Databases play a crucial role in emotional speech recognition research. These databases contain speech samples of different emotional states that are used to train and evaluate models. Some of the major emotion language databases are briefly introduced in this section.

The EmoDB dataset is a German spoken-word database created at the Technical University of Berlin [16]. It contains speech clips spoken by five female and five male actors and labeled with different emotional states: neutral, happiness, sadness, anger, fear, and disgust. EmoDB is German utterance data that can be used to consider emotional features in different languages.

RAVDESS is a database developed at Ryerson University in Canada [17]. It contains recordings of 24 North American English-speaking performers (12 male, 12 female) speaking or singing given sentences in specific emotional states. The database includes different emotional states, such as neutral, happy, sad, angry, surprise, fearful, disgust, and calm, in the emotion labels. RAVDESS provides utterances with different emotional intensity for each performer, which is useful for exploring the diversity of emotional expression.

SAVEE (Surrey Audio-Visual Expressed Emotion) is a database created at the University of Surrey in the UK that captures different emotional states through the utterances of four male actors [27]. The emotion labels include different emotional states, such as neutral, happiness, sadness, anger, surprise, fear, and disgust.

These emotional speech databases are an important source for researchers to analyze different emotional expressions and speech styles, and for model training and evaluation.

In this study, we propose a method to improve the performance of emotional speech recognition using RAVDESS and EmoDB databases.

## 3. Proposed Method

### *3.1. Data Preprocessing*

This study utilizes RAVDESS and EmoDB emotion speech databases. Although both contain various emotional states, this research concentrates on anger, disgust, fear, happiness, neutral, and sadness. Table 1 summarizes the quantity of speech data for each emotional state in each database.

**Table 1.** Data distribution by emotion in RAVDESS and EmoDB datasets.

| **Emotion** | **RAVDESS** | **EmoDB** |
|---|---|---|
| Angry (Anger) | 192 | 127 |
| Disgust (Disgust) | 192 | 46 |
| Fearful (Fear) | 192 | 69 |
| Happy (Happiness) | 192 | 71 |
| Neutral (Neutral) | 96 | 79 |
| Sad (Sadness) | 192 | 62 |
| Total | 1440 | 454 |

One of the crucial stages in utilizing speech data is data preprocessing. This process targets the removal of extraneous components and the transformation of data into a form that satisfies the requisites of the model, all while retaining the distinctive features of the speech data. We first employ envelope detection to eliminate silent and redundant segments of the speech data. Envelope detection in the librosa package proves helpful for identifying the primary variations in an audio signal and efficiently eliminating silence [28]. The process involves using Short-Time Fourier Transform (STFT) [29] to divide the audio signal into multiple frames and locate the maximum amplitude in each frame. These maximums are concatenated to form an envelope, which can then be utilized to isolate and eliminate the silent portions from the original audio signal. As a result, the data are preprocessed to remove extraneous information and retain only essential audio data. Figure 1 illustrates a comparison between the original speech and the speech with silent parts eliminated via envelope detection.

The data were converted into a mel-spectrogram after detecting the envelope, utilizing the mel-spectrogram function from the librosa package. Subsequently, the mel-spectrogram was transformed to a dB scale via Equation (4) for more consistent and efficient model training, which reduced the dynamic range of the mel-spectrogram.

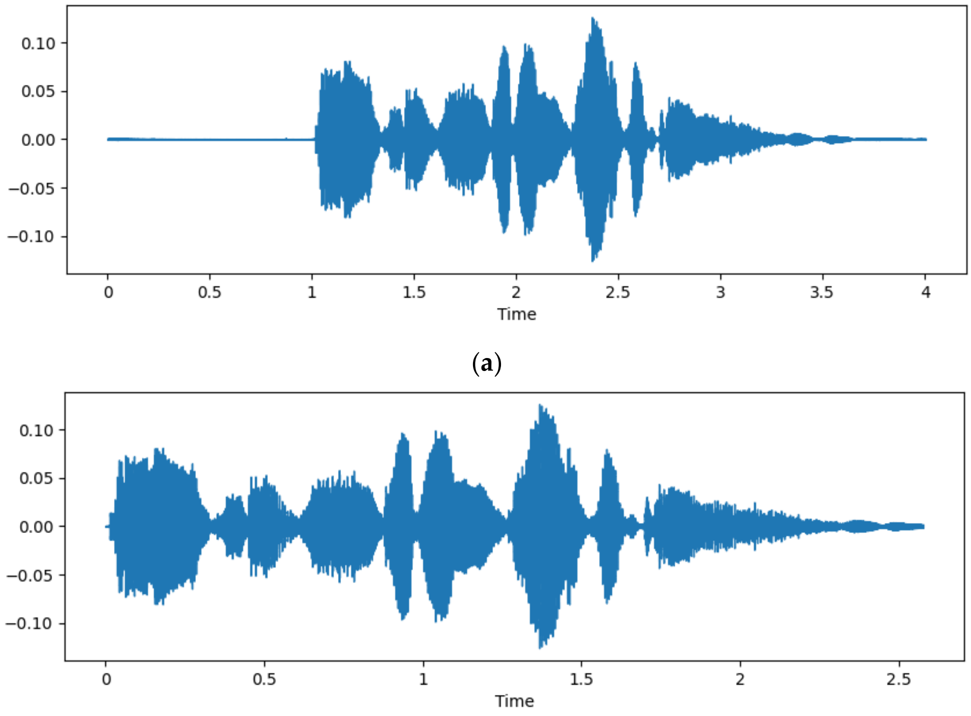
$$S_{dB} = 10 * \log_{10}\left(\frac{S}{ref}\right) \tag{4}$$

The "ref" is set to the maximum value of the mel-spectrogram. To compress the large dynamic range common in real-world environments and facilitate model training, a dB scale is applied to the mel-spectrogram. Figure 2 displays a mel-spectrogram that has been processed in this manner.

### *3.2. Mel-Spectrogram Data Augmentation Using DCGAN*

In this study, we utilized DCGAN to enhance speech data. The model underwent training through mel-spectrograms obtained from the original speech data. The trained generator resulted in fresh mel-spectrograms. A mini-batch technique was utilized due to memory limitations during deep learning training. The PyTorch deep learning framework was used to construct the model's layers.

The generator receives a random noise vector in latent space and transforms it into image-like data. In our model, we employ an initial fully connected linear layer to expand the latent vectors into 2D tensors. Subsequently, we use four transposed convolution layers to incrementally enhance the image resolution, thus yielding the final image. Batch normalization and ReLU activation functions are applied after each transposed convolution layer to ensure network stability. The generator's final layer uses the tanh activation function to confine the output within the range of [–1, 1].



(**a**)

(**b**)

**Figure 1.** (**a**) Original speech waveform; (**b**) speech waveform after envelope detection.

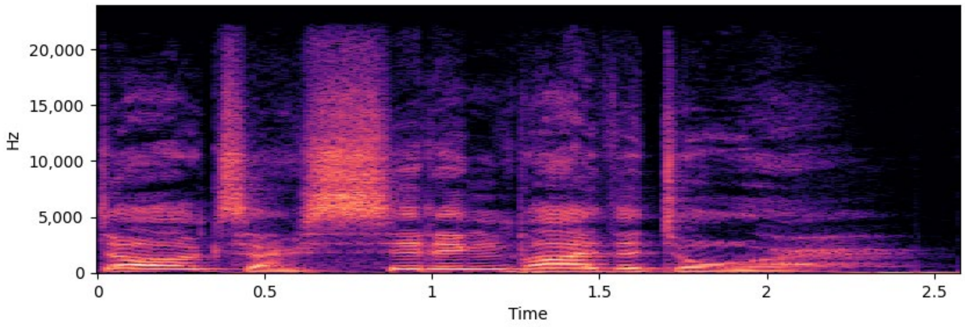


**Figure 2.** Mel-spectrogram after dB scaling.

The discriminator is a model that takes in image data to classify whether an image is genuine or generated by a generator. The framework for the discriminator comprises four convolutional layers, each incorporating batch normalization and a Leaky ReLU activation function. The last convolutional layer generates a single value, indicating the probability of an image being authentic. The sigmoid activation function then outputs the probability value within the range of [0, 1]. Table 2 shows the overall design structure for both networks.

**Table 2.** Architectures of the generator and discriminator for DCGAN-based mel-spectrogram augmentation.

| Network | Layer | Input Shape | Stride | Output Shape |
|---|---|---|---|---|
| Generator | Linear | [100] | - | [512, 8, 8] |
| | ConvTranspose2d | [512, 8, 8] | 2 | [256, 16, 16] |
| | BatchNorm2d | [256, 16, 16] | - | [256, 16, 16] |
| | ReLU | [256, 16, 16] | - | [256, 16, 16] |
| | ConvTranspose2d | [256, 16, 16] | 2 | [128, 32, 32] |
| | BatchNorm2d | [128, 32, 32] | - | [128, 32, 32] |
| | ReLU | [128, 32, 32] | - | [128, 32, 32] |
| | ConvTranspose2d | [128, 32, 32] | 2 | [64, 64, 64] |
| | BatchNorm2d | [64, 64, 64] | - | [64, 64, 64] |
| | ReLU | [64, 64, 64] | - | [64, 64, 64] |
| | ConvTranspose2d | [64, 64, 64] | 2 | [1, 128, 128] |
| | Tanh | [1, 128, 128] | - | [1, 128, 128] |
| Discriminator | Conv2d | [1, 128, 128] | 2 | [64, 64, 64] |
| | LeakyReLU | [64, 64, 64] | - | [64, 64, 64] |
| | BatchNorm2d | [64, 64, 64] | - | [64, 64, 64] |
| | Conv2d | [64, 64, 64] | 2 | [128, 32, 32] |
| | LeakyReLU | [128, 32, 32] | - | [128, 32, 32] |
| | BatchNorm2d | [128, 32, 32] | - | [128, 32, 32] |
| | Conv2d | [128, 32, 32] | 2 | [256, 16, 16] |
| | LeakyReLU | [256, 16, 16] | - | [256, 16, 16] |
| | BatchNorm2d | [256, 16, 16] | - | [256, 16, 16] |
| | Conv2d | [256, 16, 16] | 2 | [1, 8, 8] |
| | Sigmoid | [1, 8, 8] | - | [1, 8, 8] |

The mel-spectrogram data produced by this process are illustrated in Figure 3. This information was then combined with the original data under the label “fake”. This was used to train the emotion recognition model.

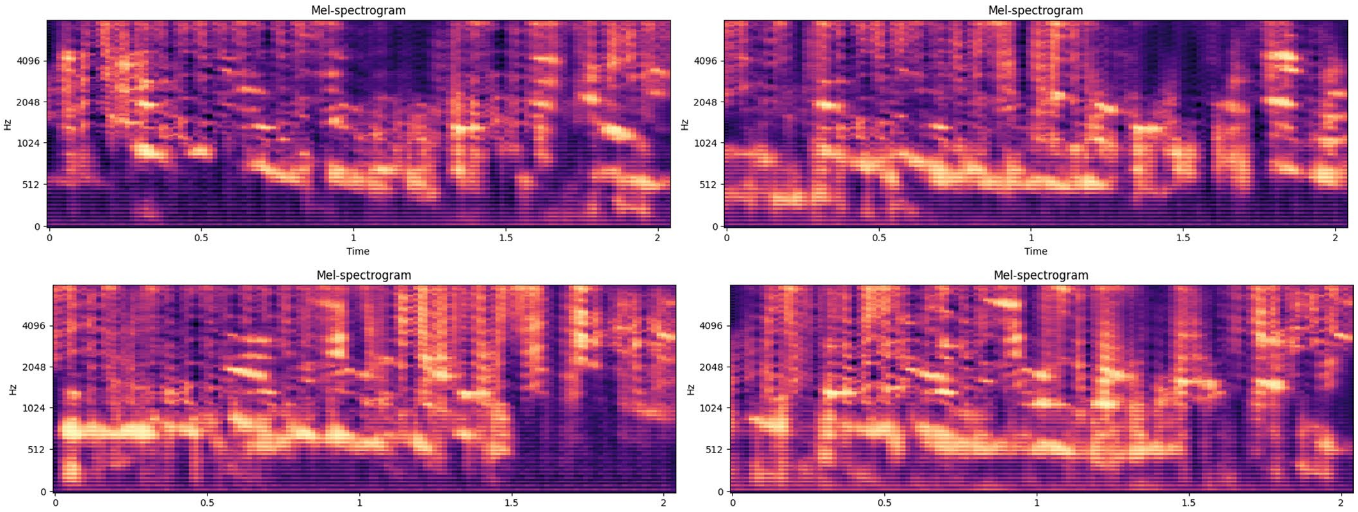


**Figure 3.** Generated mel-spectrogram using DCGAN.

### 3.3. Model Architecture: CNN-BiLSTM Emotional Speech Recognition

In this study, the original mel-spectrogram extracted from the original speech data and the mel-spectrogram generated with DCGAN were combined to form the final dataset. This dataset was used as input to a combined emotional speech recognition model of CNN and BiLSTM. The structure of the model is as follows: The first CNN module passes through a convolutional layer with 64 3 × 3 filters, applies batch normalization and ReLU activation function, and performs 2 × 2 max pooling. Next, it undergoes convolution with 128 3 × 3 filters, followed by batch normalization and an ReLU activation function. It then undergoes

4 × 4 maximum pooling. The third CNN module conducts convolution using 256 4 × 4 filters, implements batch regularization and ReLU activation functions, and executes 4 × 4 maximum pooling. To avoid overfitting, each module was subjected to drop-out. The outcome of the CNN module is transformed into the input of the LSTM and is passed through a BiLSTM layer containing 256 LSTM units. Finally, the output passes through a dense layer consisting of 128 units that incorporate L2 regularization. This is followed by a dense layer utilizing the softmax activation function, which produces the final output and denotes the probability of the class.

## 4. Experiment

### *4.1. Experimental Setting*

In our experiments, we used authentic mel-spectrograms extracted from the RAVESS and EmoDB databases as well as augmented mel-spectrograms generated using DCGAN as datasets. The model structure was developed through a combination of CNN and BiLSTM, and we opted for the RMSprop [30] optimizer to ensure stable gradient updates and rapid convergence. We set the ratio of dividing the data into train, test, and validation sets to 7:1.5:1.5 to maintain the stability of the model while performing sufficient training and evaluation. We used the ReduceLROnPlateau method to dynamically adjust the learning rate to maintain the stability of the optimization process: the initial learning rate was set to 0.001, and as the training progressed, we were able to achieve better model performance by reducing the learning rate when performance improvement was no longer observed.

The main purpose of the performance evaluation is to see how effective data augmentation with DCGAN is. For this study, we compared the performance of the original data and the augmented data combined with the original data for each dataset in RAVDESS and EmoDB. Weighted accuracy (WA) and Unweighted accuracy (UA) were used as performance measures, which are commonly applied in speech emotion recognition, especially when there is an imbalanced data distribution for each emotion class. WA is a technique that measures overall accuracy by assigning weights to each class based on its significance or frequency, which mirrors the distribution of each class in the dataset. It evaluates the accuracy of each class independently. Conversely, UA calculates the average accuracy of all classes equally, without taking data imbalance into account, and evaluates the accuracy of each class independently.

### *4.2. Result*

In this experiment, we assessed performance through the evaluation of two datasets: RAVDESS and EmoDB. To compare results, we examined the performance of using only original data versus that of incorporating augmented data for each dataset, resulting in a total of four different data configurations. To compare results, we examined the performance of using only original data versus that of incorporating augmented data for each dataset, resulting in a total of four different data configurations. Table 3 summarizes the results.

**Table 3.** Comparison of WA and UA for original and augmented datasets.

| Dataset | WA | UA |
|---|---|---|
| RAVDESS | 64.8% | 64.2% |
| RAVDESS+augmented | 72.3% | 72.3% |
| EmoDB | 80.6% | 82.6% |
| EmoDB+augmented | 90.4% | 91.3% |

In the RAVDESS dataset, solely utilizing the original data resulted in a weighted accuracy (WA) of 64.8% and an unweighted accuracy (UA) of 64.2%. However, when the augmented data were incorporated, there was a marked improvement, registering 72.3% for both WA and UA. Similarly, for the EmoDB dataset, the original data yielded a

WA of 80.6% and a UA of 82.6%, while incorporating augmented data led to a WA and UA of 90.4% and 91.3%, respectively. These results demonstrate that the performance of the speech emotion recognition model improves when utilizing the DCGAN-based data augmentation technique. Tables 4–7 present the experimental findings, specifically focusing on the UA value in the confusion matrix. The colored numbers in the table footer are the highest scores in each emotion group.

**Table 4.** Confusion matrix for RAVDESS dataset (%).

| | | **Predicted** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Angry** | **Disgust** | **Fear** | **Happy** | **Neutral** | **Sad** |
| True | angry | 75.9 | 13.8 | 3.4 | 3.4 | 3.4 | 0 |
| | disgust | 6.9 | 72.4 | 0 | 6.9 | 0 | 13.8 |
| | fear | 3.4 | 3.4 | 51.7 | 6.9 | 0 | 34.5 |
| | happy | 17.2 | 10.3 | 6.9 | 41.4 | 0 | 24.1 |
| | neutral | 0 | 7.1 | 7.1 | 0 | 71.4 | 14.3 |
| | sad | 3.4 | 6.9 | 10.3 | 0 | 3.4 | 75.9 |

**Table 5.** Confusion matrix for RAVDESS+augmented dataset (%).

| | | **Predicted** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Angry** | **Disgust** | **Fear** | **Happy** | **Neutral** | **Sad** |
| True | angry | 82.8 | 6.9 | 6.9 | 3.4 | 0 | 0 |
| | disgust | 0 | 79.3 | 0 | 6.9 | 0 | 13.8 |
| | fear | 3.4 | 6.9 | 65.5 | 0 | 6.9 | 17.2 |
| | happy | 10.3 | 6.9 | 6.9 | 51.7 | 6.9 | 17.2 |
| | neutral | 7.1 | 14.3 | 7.1 | 0 | 71.4 | 0 |
| | sad | 0 | 3.4 | 6.9 | 3.4 | 3.4 | 82.8 |

**Table 6.** Confusion matrix for EmoDB dataset (%).

| | | **Predicted** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Angry** | **Disgust** | **Fear** | **Happy** | **Neutral** | **Sad** |
| True | angry | 94.7 | 0 | 5.3 | 0 | 0 | 0 |
| | disgust | 14.3 | 71.4 | 0 | 0 | 0 | 14.3 |
| | fear | 0 | 9.1 | 81.8 | 0 | 0 | 9.1 |
| | happy | 18.2 | 0 | 9.1 | 63.6 | 9.1 | 0 |
| | neutral | 0 | 0 | 8.3 | 0 | 83.3 | 8.3 |
| | sad | 0 | 0 | 0 | 0 | 11.1 | 88.9 |

**Table 7.** Confusion matrix for EmoDB+augmented dataset (%).

| | | **Predicted** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Angry** | **Disgust** | **Fear** | **Happy** | **Neutral** | **Sad** |
| True | angry | 94.7 | 0 | 0 | 5.3 | 0 | 0 |
| | disgust | 0 | 85.7 | 0 | 0 | 0 | 14.3 |
| | fear | 0 | 0 | 90.9 | 0 | 0 | 9.1 |
| | happy | 18.2 | 0 | 0 | 81.8 | 0 | 0 |
| | neutral | 0 | 0 | 0 | 0 | 100 | 0 |
| | sad | 0 | 0 | 11.1 | 0 | 0 | 88.9 |

## 5. Conclusions

In this study, we proposed a method to augment emotional speech data using DCGAN. Using the proposed method, a speech emotion recognition model was trained using the original data along with the augmented mel-spectrogram data generated from the RAVDESS and EmoDB datasets. The experiments indicate that the inclusion of DCGAN-generated data in the training set leads to significant improvements in model performance as compared with only using the original data. In our experiments, we also evaluated the performance of the model using two major performance evaluation metrics, WA and UA.

This research demonstrates that using generative models like DCGAN for data augmentation is an effective approach to construct high-performance models for speech emotion recognition, especially when the size of the emotional speech dataset is limited. In future work, we will further verify the generality of the proposed method by utilizing different generation models and different speech datasets. Furthermore, we believe that a deeper study of the characterization of the augmented data and the resulting performance changes in the speech emotion recognition model is necessary.

**Author Contributions:** Conceptualization, J.-Y.B. and S.-P.L.; methodology, J.-Y.B.; investigation, J.-Y.B.; writing—original draft preparation, J.-Y.B.; writing—review and editing, S.-P.L.; project administration, S.-P.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the 2023 Research Grant from Sangmyung University.

**Data Availability Statement:** Experiments used publicly available datasets.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| BiLSTM | Bidirectional long short-term memory; |
| CNN | Convolutional neural network; |
| DCGAN | Deep convolutional generative adversarial network; |
| GAN | Generative adversarial network; |
| GMM | Gaussian mixture model; |
| HMM | Hidden Markov model; |
| NLP | Natural language processing; |
| ReLU | Rectified linear unit; |
| STFT | Short-time Fourier transform; |
| WA | Weighted accuracy; |
| UA | Unweighted accuracy |

## References

1. Schuller, B.; Rigoll, G.; Lang, M. Hidden Markov model-based speech emotion recognition. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03), Hong Kong, China, 6–10 April 2003; IEEE: Piscataway, NJ, USA, 2003; Volume 2, p. I-401. [CrossRef]
2. Nogueiras, A.; Moreno, A.; Bonafonte, A.; Mariño, J.B. Speech emotion recognition using hidden Markov models. In Proceedings of the Seventh European Conference on Speech Communication and Technology, Aalborg, Denmark, 3–7 September 2001. [CrossRef]
3. Lin, Y.-L.; Wei, G. Speech emotion recognition based on HMM and SVM. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 8, pp. 4898–4901. [CrossRef]
4. Lanjewar, R.B.; Mathurkar, S.; Patel, N. Implementation and comparison of speech emotion recognition system using Gaussian Mixture Model (GMM) and K-Nearest Neighbor (K-NN) techniques. *Procedia Comput. Sci.* **2015**, *49*, 50–57. [CrossRef]
5. Hu, H.; Xu, M.-X.; Wu, W. GMM supervector based SVM with spectral features for speech emotion recognition. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, Honolulu, HI, USA, 16–20 April 2007; IEEE: Piscataway, NJ, USA, 2007; Volume 4, pp. IV-413–IV-416. [CrossRef]
6. Abbaschian, B.J.; Sierra-Sosa, D.; Elmaghraby, A. Deep learning techniques for speech emotion recognition, from databases to models. *Sensors* **2021**, *21*, 1249. [CrossRef] [PubMed]
7. Issa, D.; Demirci, M.F.; Yazici, A. Speech emotion recognition with deep convolutional neural networks. *Biomed. Signal Process. Control* **2020**, *59*, 101894. [CrossRef]
8. Meng, H.; Yan, T.; Yuan, F.; Wei, H. Speech emotion recognition from 3D log-mel spectrograms with deep learning network. *IEEE Access* **2019**, *7*, 125868–125881. [CrossRef]
9. Makhmudov, F.; Kutlimuratov, A.; Akhmedov, F.; Abdallah, M.S.; Cho, Y.-I. Modeling Speech Emotion Recognition via Attention-Oriented Parallel CNN Encoders. *Electronics* **2022**, *11*, 4047. [CrossRef]
10. Mustaqeem; Kwon, S. A CNN-assisted enhanced audio signal processing for speech emotion recognition. *Sensors* **2019**, *20*, 183. [CrossRef] [PubMed]

11. Abdelhamid, A.A.; El-Kenawy, E.-S.M.; Alotaibi, B.; Amer, G.M.; Abdelkader, M.Y.; Ibrahim, A.; Eid, M.M. Robust speech emotion recognition using CNN+ LSTM based on stochastic fractal search optimization algorithm. *IEEE Access* **2022**, *10*, 49265–49284. [CrossRef]
12. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical report; University of Toronto: Toronto, ON, Canada, 2009.
13. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255. [CrossRef]
14. LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 31 August 2023).
15. Busso, C.; Bulut, M.; Lee, C.-C.; Kazemzadeh, A.; Mower, E.; Kim, S.; Chang, J.N.; Lee, S.; Narayanan, S.S. IEMOCAP: Interactive emotional dyadic motion capture database. *Lang. Resour. Eval.* **2008**, *42*, 335–359. [CrossRef]
16. Burkhardt, F.; Paeschke, A.; Rolfes, M.; Sendlmeier, W.F.; Weiss, B. A database of German emotional speech. In Proceedings of the 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, 4–8 September 2005.
17. Livingstone, S.R.; Russo, F.A. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLoS ONE* **2018**, *13*, e0196391. [CrossRef] [PubMed]
18. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**. [CrossRef]
19. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
20. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9. [CrossRef]
21. Zeiler, M.D.; Taylor, G.W.; Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In Proceedings of the 2011 International Conference on Computer Vision, Washington, DC, USA, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 2018–2025. [CrossRef]
22. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 448–456.
23. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
24. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Atlanta, GA, USA, 16 June 2013; p. 3.
25. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
27. Jackson, P.; Haq, S. *Surrey Audio-Visual Expressed Emotion (Savee) Database*; University of Surrey: Guildford, UK, 2014.
28. Librosa. Available online: https://librosa.org (accessed on 13 August 2023).
29. Allen, J. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE Trans. Acoust. Speech Signal Process.* **1977**, *25*, 235–238. [CrossRef]
30. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.