CNN-RNN and Data Augmentation Using Deep Convolutional Generative Adversarial Network for Environmental Sound Classification

Behnaz Bahmei, Elina Birmingham, and Siamak Arzanpour, *Member, IEEE*

***Abstract*—Deep neural networks in deep learning have been widely demonstrated to have higher accuracy and distinct advantages over traditional machine learning methods in extracting data features. While convolutional neural networks (CNNs) have shown great success in feature extraction and audio classification, it is important to note that real-time audios are dependent on previous scenes. Also, the main drawback of deep learning algorithms is that they need a huge number of datasets to indicate their efficient performance. In this paper, a recurrent neural network (RNN) combined with CNN is proposed to address this problem. Moreover, a Deep Convolutional Generative Adversarial Network (DCGAN) is used for high-quality data augmentation. This data augmentation technique is applied to the UrbanSound8K dataset to improve the environmental sound classification. Batch normalization, transfer learning, and three feature representations map are used to improve the model accuracy. The results show that the generated images by DCGAN have similar features to the original training images and has the capability to generate spectrograms and improve the classification accuracy. Experimental results on UrbanSound8K datasets demonstrate that the proposed CNN-RNN architecture achieves better performance than the state-of-the-art classification models.**

***Index Terms*—Data augmentation, deep convolutional generative adversarial networks, environmental sound classification, convolutional recurrent neural network (CRNN).**

I. Introduction

In recent years, there has been significant research activity in the area of developing intelligent sound recognition systems that are able to accurately classify different types of environmental sounds. There is a wide range of potential applications of environmental sound classification (ESC), including assistive technologies (like tools for hearing impaired [1]), context awareness [2], surveillance [3], urban planning [4], biology [5], and monitoring [6].

In the ESC problem, the goal is to recognize a specific sound source, such as dog barking, siren, and drilling. These sounds include various audio events with chaotic and diverse structure and can be categorized into three groups including, single sounds such as a mouse-click, repeated discrete sound such as clapping hands or typing on a keyboard, and steady continuous sounds such as the sound of a vacuum cleaner [7].

There exist a various number of machine learning algorithms to improve audio classification accuracy like Support Vector Machine (SVM) [8], hidden Markov models (HMM) [9], gaussian mixture models (GMM) [10], k-nearest neighbor (KNN) algorithm [11]. In recent years, deep learning techniques have been introduced to enhance the recognition performance of environmental sounds [12], [13]. Due to the learning capability of the hierarchical features from high-dimensional raw data, deep neural networks in deep learning are more accurate than the traditional techniques [12]. For example, Convolutional Neural Networks (CNNs) have been actively used for various sound classification tasks [13], [14] and have shown promising performance. Also, Recurrent Neural Networks (RNNs) are capable of modelling sequential data such as videos and word sequences [15], [16]. Recently, these structures have been combined in order to take the advantages of both RNNs and CNNs. This structure was first proposed in [17] for document classification and later applied to image classification [18], music transcription [19] and audio classification [20]. These studies have shown that the CNN-RNN framework can learn better image feature learning and achieves superior performance than the state-of-the-art methods.

Deep learning algorithms need a huge amount of data for efficient performance. As a result, the main challenge associated with the deep learning problem is to provide an appropriate amount of data to train the network. Data annotation is the most challenging part of training deep learning supervised models. It is a time-consuming procedure that requires considerable effort and cost, especially when the network needs large number of samples. There are various data augmentation techniques including, flip, rotation, scale, crop, translation, and Gaussian noise [13], [21]. These augmentations, however, are mostly being implemented with low-level transformations, which in general are not capable of improving the performance of conventional or advanced deep learning classifiers. Recently, generative adversarial networks (GAN) [22] have created new opportunities for researchers to generate new and high-quality results similar to the existing data samples. The idea of GANs is to simultaneously train two models, a generative model G, and discriminative model D. The generative model creates photorealistic images

Manuscript received November 5, 2021; revised January 28, 2022; accepted January 31, 2022. Date of publication February 21, 2022; date of current version March 10, 2022. This work was supported by Kids Brain Health Network (KBHN). The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Ville M. Hautamaki. *(Corresponding author: Siamak Arzanpour.)*

Behnaz Bahmei and Siamak Arzanpour are with the School of Mechatronics System Engineering, Simon Fraser University, Surrey, BC V3T 0A3, Canada (e-mail: bbahmei@sfu.ca; arzanpour@sfu.ca).

Elina Birmingham is with the Faculty of Education, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: elina_birmingham@sfu.ca).

Digital Object Identifier 10.1109/LSP.2022.3150258

that are similar to the original data distribution and the task of discriminative model is to determine whether a given image looks realistic (image came from the training data) or looks like it has been artificially created by the generative model. However, because of the battle of two networks, GANs are known to have several pitfalls. More specifically, they are unstable to train, i.e., the outputs that are produced by the generative model are often inaccurate. In addition, GANs are hard to tune and get to work properly. To tackle those drawbacks, a class of GANs called Deep Convolutional Generative Adversarial Networks (DCGAN) is proposed that has a set of architectural constraints to stabilize GANs [23].

This paper aims to improve the environmental sound classification process by using a unified CNN-RNN framework for classification and a DCGAN framework for high-quality data augmentation. To summarize, the main contributions of this paper are as follows:

1) A DCGAN structure is used to generate high scalability and excellent data samples. This structure is evaluated on the UrbanSound8K dataset [24]. The experimental results show that the model can generate data samples that have similar structures.
2) A unified CNN-RNN classification framework is designed and trained on original and generated samples to achieve a very high-level of classification accuracy.

The composition of the paper is as the following: In Section II, the methods including classification, feature extraction, transfer learning [25], and data augmentation are discussed. In Section III, the experimental setting for data augmentation and classification processes are explained. Then, Section IV provides details about the experimental results. Finally, the conclusions are presented in Section V.

II. Methods

In this section, the methods for classification, feature extraction, and data augmentation are explained.

A. Classification

An RNN is a class of artificial neural networks in which the connection lines between the nodes are sequential. It allows RNN to model the dynamic temporal behavior of sequences through directed cyclic connections between the nodes [26]. RNN models have various structures including Long Short-Term Memory (LSTM) [27], Gated Recurrent Unit (GRU) [28], Initialized Recurrent Neural Networks (IRNN) [29] and Convolutional LSTM Network (Conv-LSTM) [30]. In this paper Deep GRU-based RNN architecture has been used since the GRU controls the flow of the information and unlike the LSTM, it does not need to use a memory unit. Moreover, the GRU is computationally efficient and has better performance than LSTM [31].

CNN has been studied recently and used as an automated extractor of features [13]. Generally speaking, a feature extractor takes an input image, extracts features from the input, and creates an output vector. Convolutional layers at the core of the CNN apply sliding filters across the image height and width to automatically extract the features. By multiplying pixel values and filter results which are learned across multiple epochs, the final map is produced. The CNN part in this framework extracts semantic representations from images to feed the RNN part.

A CNN-RNN can be described as a combination of CNN and RNN frameworks. In other words, CNN plays the feature extractor role and RNN plays the temporal summarizer role. Adopting an RNN for aggregating the features enables the networks to take the global structure into account while local features are extracted by the remaining convolutional layers.

B. Feature Extraction and Transfer Learning

There exist different feature extraction techniques for audio and speech recognition including zero-crossing rate (ZCR) [32], short-time energy (STE) [33], Mel Frequency Coefficients (MFCC). The Cepstrum MFCC is a logarithmic frequency scale and is considered as one of the most common and effective features for speech and audio recognition [13]. The main idea of MFCC feature extraction is to develop a scale that adapts well to human hearing. The MFCC feature connects the perceived frequency, or pitch, of a sound to its actual measured frequency.

In the present paper, a total of three feature engineering techniques are used to build three feature representation maps including mel-spectrogram, and decomposing audio time-series data to harmonic and percussive components. They ultimately provide a three-dimensional image feature map for each audio. After creating the feature maps, transfer learning is used as a method to extract features from the feature maps. Transfer learning is a strong tool that recently is used in machine learning projects to utilize knowledge from previously learned models and apply them to newer related ones. In this paper, the exceptional capability of transfer learning is adopted by using the pre-trained VGG-19 (Visual Geometry Group) [34] model as a feature extractor to extract bottleneck features from all images. VGG-19 is a convolutional neural network model trained on the ImageNet dataset which contains over 14 million images belonging to 1000 classes.

C. Data Augmentation

Data augmentation is a powerful strategy to increase the diversity of available data and make it possible to train models without collecting new data. In this paper, two data augmentation strategies including traditional and intelligent augmentation are considered. In traditional data augmentation, two different audio data deformations are used. First, some background noises (crowd sound, street sound, restaurant) added to the data samples (the background noises have been extracted from public recordings available through the “freesound.org” website [35]). Second, pitch shifting [13] is applied to the data. The pitch of the audio samples is changed by a half octave (up and down) to create different sounds. Each deformation is applied directly to the audio signal before it is converted to the input representation.

For the intelligent data augmentation, a DCGAN structure is used. The GAN framework consists of two models including a generative model and a discriminative model. The generator creates samples close to the training samples and feeds them to the discriminator. The discriminator then receives the images from the generator and also the real images to compare them and

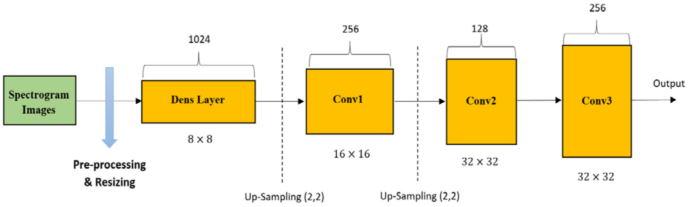


Fig. 1. The DCGAN generator Architecture.

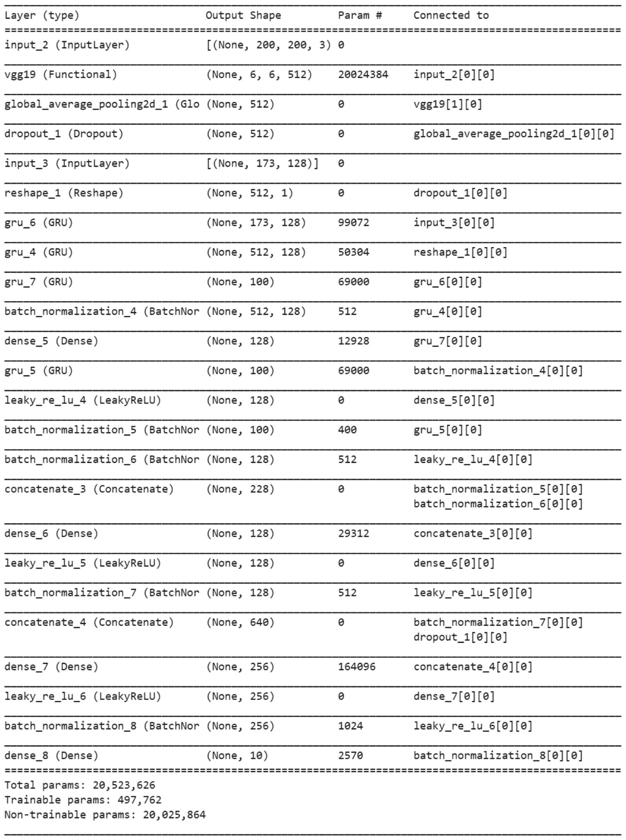
distinguish real ones. Discriminator (D) is a set of convolution layers with strided convolutions, so it downsamples the input image at every convolution layer. On the other hand, generator (G) is a set of convolution layers with fractional-strided convolutions or transpose convolutions, so it upsamples the input image at every convolution layer [23].

D. Experimental Setup

In this section, the performance of the DCGAN is examined using the UrbanSound8K dataset to generate further data samples and improve the performance of the classification model for environmental sound recognition. Subsequently, a united CNN-RNN framework is designed for the feature extraction and classification process. The length of each data sample is up to 5 seconds. MFCC feature extraction is used, and the mel-spectrograms are generated. The dimension of the original images is 768 × 384. To avoid a huge amount of training parameters in the DCGAN training process, the original images are resized to 64 × 64, where the total number of frames (columns) is 64 and the total number of bands (rows) is 64.

The architectures for the generators and discriminators are illustrated in Fig. 1. The DCGAN architecture in this study is comprised of 3 convolutional layers and 1 dense layer. Batch normalization is applied to every layer of the network to learn more efficiently. The model is trained over 4000 epochs using a batch size of 32. The DCGAN architecture hyperparameters are set based on [23]. In the generator model, the ReLU activation function is used after each layer except the last one. For the last layer, the hyper tangent activation function is applied to obtain the image of 3 channels. The slope on the leaky ReLU is 0.2. For the discriminator, instead of a hyper tangent, standard sigmoid activation is used on the output layer to determine the probability of the generated image. There is no batch normalization on the first layer, and also no pooling layers for down-sampling. The stride size for the discriminator is 4, 4, 4, 2 respectively. The ADAM optimizer [36] is used in order to update network weights.

The CNN-RNN framework is shown in Fig. 2. The CNN part extracts semantic representations from the inputs. The RNN part models label relationship and label dependency. To train the networks, Adam, categorical cross-entropy, leaky ReLU and SoftMax are used as an optimizer, loss function, and activation functions in the output layers, respectively. To reduce overfitting and improve generalization error, two methods are used. The Batch Normalization technique is imposed on RNN weights at the output of the RNN layer and dropout layers are used to randomly drop out the nodes during the training.



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 200, 200, 3) 0
__________________________________________________________________________________________________
vgg19 (Functional)              (None, 6, 6, 512)    20024384    input_2[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_1 (Glo (None, 512)          0           vgg19[1][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 512)          0           global_average_pooling2d_1[0][0]
__________________________________________________________________________________________________
input_3 (InputLayer)            [(None, 173, 128)]   0
__________________________________________________________________________________________________
reshape_1 (Reshape)             (None, 512, 1)       0           dropout_1[0][0]
__________________________________________________________________________________________________
gru_6 (GRU)                     (None, 173, 128)     99072       input_3[0][0]
__________________________________________________________________________________________________
gru_4 (GRU)                     (None, 512, 128)     50304       reshape_1[0][0]
__________________________________________________________________________________________________
gru_7 (GRU)                     (None, 100)          69000       gru_6[0][0]
__________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 512, 128)     512         gru_4[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                 (None, 128)          12928       gru_7[0][0]
__________________________________________________________________________________________________
gru_5 (GRU)                     (None, 100)          69000       batch_normalization_4[0][0]
__________________________________________________________________________________________________
leaky_re_lu_4 (LeakyReLU)       (None, 128)          0           dense_5[0][0]
__________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 100)          400         gru_5[0][0]
__________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 128)          512         leaky_re_lu_4[0][0]
__________________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 228)          0           batch_normalization_5[0][0]
                                                                 batch_normalization_6[0][0]
__________________________________________________________________________________________________
dense_6 (Dense)                 (None, 128)          29312       concatenate_3[0][0]
__________________________________________________________________________________________________
leaky_re_lu_5 (LeakyReLU)       (None, 128)          0           dense_6[0][0]
__________________________________________________________________________________________________
batch_normalization_7 (BatchNor (None, 128)          512         leaky_re_lu_5[0][0]
__________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 640)          0           batch_normalization_7[0][0]
                                                                 dropout_1[0][0]
__________________________________________________________________________________________________
dense_7 (Dense)                 (None, 256)          164096      concatenate_4[0][0]
__________________________________________________________________________________________________
leaky_re_lu_6 (LeakyReLU)       (None, 256)          0           dense_7[0][0]
__________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, 256)          1024        leaky_re_lu_6[0][0]
__________________________________________________________________________________________________
dense_8 (Dense)                 (None, 10)           2570        batch_normalization_8[0][0]
==================================================================================================
Total params: 20,523,626
Trainable params: 497,762
Non-trainable params: 20,025,864
__________________________________________________________________________________________________
```

Fig. 2. The proposed architecture of the CNN-RNN model for classification.

10-fold cross validation is used to evaluate the model's performance. According to Fig. 3, the model structure contains two directions which are joined together at a concatenate layer. In one branch, the CNN part extracts semantic representations from the three-dimensional feature map that are prepared by the methods explained in Section II (features extraction). All the Input data including the generated samples and the original samples are reshaped to (200, 200, 3) and fed to the CNN layers as the input. On the other side, to benefit from RNN features, the second branch is considered using GRU layers. Input data to this branch is a 2D mel-spectrogram with the shape of (time steps = 173, number of mels = 128). These two branches are eventually concatenated to create the final layer of the feature extraction.

III. Results and Discussion

The efficiency of the proposed DCGAN approach and the united CNN-RNN framework in this paper is evaluated on the UrbanSound8K dataset.

Fig. 3 illustrates random visual examples of the generated spectrograms using DCGAN. As you can see in this figure, DCGAN has a high capability to produce spectrograms that have similar structures. Overall, the size of the datasets is increased with extra 1000 samples using DCGAN. In order to investigate that the generated images have the capability to improve the classification process, a CNN-RNN algorithm is used with a mix of the real dataset and DCGAN's generated images. As can be seen in Table I, the accuracy is increased by over 4% compared to the case using original data. It indicates that the generated

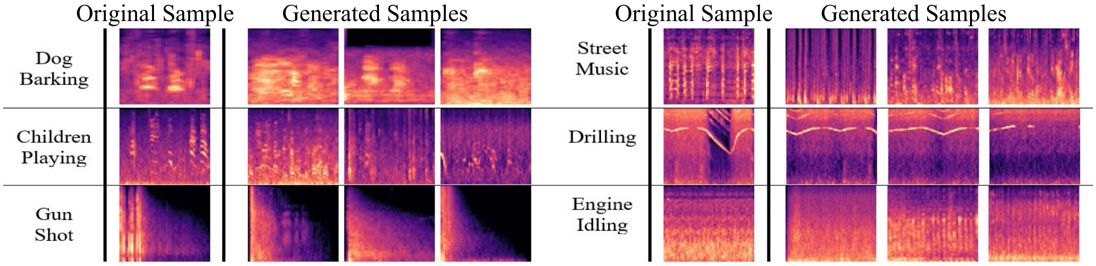


Fig. 3. Generated spectrograms using DCGAN. The first image of each row shows the original image images.

TABLE I
THE CLASSIFIER'S ACCURACY USING ORIGINAL IMAGES VS. ORIGINAL AND GENERATED IMAGES

	Classification Accuracy (%)
Original	93.3
Original + Generated	98.0

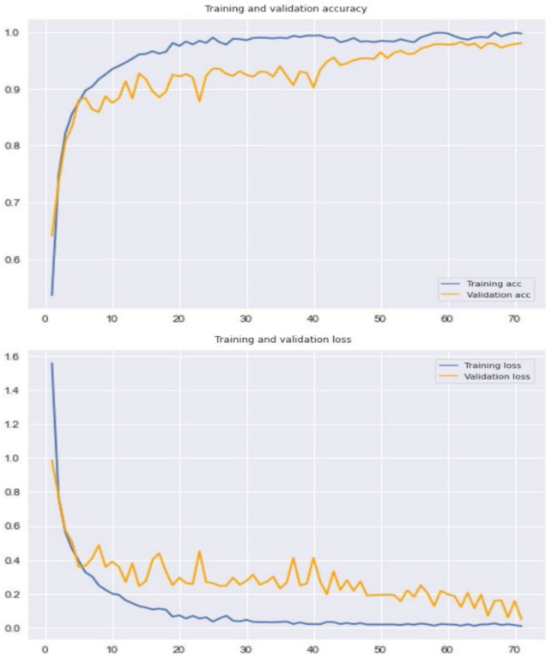


Fig. 4. Overall accuracy and loss.

TABLE II
PREVIOUS STATE-OF-THE-ART ESC MODELS VS. THE PROPOSED MODEL IN THIS PAPER ON URBANSOUND8K DATASET

Framework	Classification Accuracy (%)	Ref.
PiczakCNN	73.7	[12]
AlexNet	92	[37]
Google Net	93	[37]
RNN	82.09	[31]
MC-Net + LMC	95	[38]
WCCGAN	94	[39]
The proposed model	98	

TABLE III
AN EVALUATION OF THE PROPOSED MODEL ON ESC-50 DATASET

Framework	Classification Accuracy (%)	Ref.
PiczakCNN	64.5	[12]
AlexNet	69	[37]
Google Net	73%	[37]
The proposed model with data augmentation	91.5	
The proposed model without data augmentation	88.7	

images have similar features to the original ones for CNN. It also shows the capability of DCGAN to generate spectrograms. The overall accuracy and loss function of the models are presented in Fig. 4 to get a better understanding of the model performance. Based on the results shown in Fig. 4, the model loss and accuracy between the training and validation process is quite consistent. It can be seen that the improved accuracy of the CNN-RNN model is higher than that of the original CNN and RNN models.

Table II is provided in order to compare the accuracy of other deep learning approaches on the same dataset. From this table, it is shown that the proposed model is able to surpass other states of the art models including CNN, RNN on UrbanSound8K datasets.

In addition, ESC-50 dataset is used to further evaluate the performance of the proposed model. the ESC-50 has 50 classes but is considered more challenging because there are only 40 samples in each class. The simulation results show that even though the performance of the model on the ESC-50 is lower than UrbanSound8k due to the number of training samples, the overall performance of the model is higher compared to the other excising methods on the same dataset. Table III compares the simulation results for the proposed model on ESC-50 using the data augmentation with simulation without data augmentation, and other existing state-of-the-art articles.

IV. Conclusion

In this paper, a unified CNN-RNN framework for environmental sound classification is proposed. A generative model using DCGAN is used to addresses the lack of data problem for environmental sound classification. This data augmentation method can produce spectrograms with similar structures to the training set. Applying a CNN-RNN algorithm on a mix of the real dataset and generated images show that the DCGAN method has the ability to improve the performance of the environmental sound classification task. The CNN-RNN framework proposed in this paper combines the advantages of CNN and RNN. Experimental results on UrbanSound8K datasets demonstrate that the proposed approach achieves superior performance to the state-of-the-art methods.

REFERENCES

[1] E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras, "Feature selection for sound classification in hearing aids through restricted search driven by genetic algorithms," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2249–2256, Nov. 2007, doi: 10.1109/TASL.2007.905139.

[2] D. P. Mital and G. W. Leng, "A voice-activated robot with artificial intelligence," in *Proc. Int. Conf. Ind. Electron., Control Instrum.*, vol. 2, 1991, pp. 904–909, doi: 10.1109/IECON.1991.239170.

[3] K. Łopatka, P. Zwan, and A. Czyzewski, "Dangerous sound event recognition using support vector machine classifiers," *Adv. Intell. Soft Comput.*, vol. 80, pp. 49–57, 2010, doi: 10.1007/978-3-642-14989-4_5.

[4] D. Barchiesi, D. D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 16–34, May 2015, doi: 10.1109/MSP.2014.2326181.

[5] F. R. González-Hernández, L. P. Sánchez-Fernández, S. Suárez-Guerra, and L. A. Sánchez-Pérez, "Marine mammal sound classification based on a parallel recognition model and octave analysis," *Appl. Acoust.*, vol. 119, pp. 17–28, Apr. 2017, doi: 10.1016/J.APACOUST.2016.11.016.

[6] L. Ballan, A. Bazzica, M. Bertini, A. Del Bimbo, and G. Serra, "Deep networks for audio event classification in soccer videos," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2009, pp. 474–477, doi: 10.1109/ICME.2009.5202537.

[7] S. Li, Y. Yao, J. Hu, G. Liu, X. Yao, and J. Hu, "An ensemble stacked convolutional neural network model for environmental event sound recognition," *Appl. Sci.*, vol. 8, no. 7, Jul. 2018, Art. no. 1152, doi: 10.3390/APP8071152.

[8] S. Sameh and Z. Lachiri, "Multiclass support vector machines for environmental sounds classification in visual domain based on log-Gabor filters," *Int. J. Speech Technol.*, vol. 16, no. 2, pp. 203–213, Jun. 2013, doi: 10.1007/S10772-012-9174-0.

[9] Y. T. Peng, C. Y. Lin, M. T. Sun, and K. C. Tsai, "Healthcare audio event classification using hidden Markov models and hierarchical hidden Markov models," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2009, pp. 1218–1221, doi: 10.1109/ICME.2009.5202720.

[10] G. Shen, Q. Nguyen, and J. S. Choi, "An environmental sound source classification system based on mel-frequency cepstral coefficients and Gaussian mixture models," *IFAC Proc. Vol.*, vol. 45, no. 6, pp. 1802–1807, May 2012, doi: 10.3182/20120523-3-RO-2023.00251.

[11] J. C. Wang, J. F. Wang, K. W. He, and C. S. Hsu, "Environmental sound classification using hybrid SVM/KNN classifier and MPEG-7 audio low-level descriptor," in *Proc. IEEE Int. Joint Conf. Neural Netw. Proc.*, 2006, pp. 1731–1735, doi: 10.1109/IJCNN.2006.246644.

[12] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proc. IEEE Int. Work. Mach. Learn. Signal Process.*, Nov. 2015, pp. 1–6, doi: 10.1109/MLSP.2015.7324337.

[13] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017, doi: 10.1109/LSP.2017.2657381.

[14] S. Adapa, "Urban sound tagging using convolutional neural networks," DCASE2019 Challenge Tech. Rep. Workshop, pp. 5–9, Sep. 2019, doi: 10.33682/8axe-9243.

[15] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning sound representations from unlabeled video," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Oct. 2016, pp. 892–900.

[16] T. H. Vu and J. C. Wang, "Acoustic scene and event recognition using recurrent neural networks," *Detect. Classification Acoust. Scenes Events*, vol. 2016. pp. 1–3, 2016.

[17] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Proc. Conf. Empir. Methods Natural Lang. Process.*, 2015, pp. 1422–1432, doi: 10.18653/V1/D15-1167.

[18] Z. Zuo *et al.*, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2015, pp. 18–26, doi: 10.1109/CVPRW.2015.7301268.

[19] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 5, pp. 927–939, May 2016.

[20] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 6, pp. 1291–1303, Feb. 2017, doi: 10.1109/TASLP.2017.2690575.

[21] X. Zhu, Y. Liu, J. Li, T. Wan, and Z. Qin, "Emotion classification with data augmentation using generative adversarial networks," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, Cham, Switzerland: Springer, Jun. 2018, pp. 349–360, doi: 10.1007/978-3-319-93040-4_28.

[22] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Jun. 2014, doi: 10.1145/3422622.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Int. Conf. Learn. Represent.*, Nov. 2015, pp. 1–16. [Online]. Available: https://arxiv.org/abs/1511.06434v2

[24] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 1041–1044, doi: 10.1145/2647868.2655045.

[25] J. Lu, R. Ma, G. Liu, and Z. Qin, "Deep convolutional neural network with transfer learning for environmental sound classification," in *Proc. Int. Conf. Comput. Control Robot. (ICCCR)*, Jan. 2021, pp. 242–245, doi: 10.1109/ICCCR49711.2021.9349393.

[26] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2285–2294, doi: 10.1109/CVPR.2016.251.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[28] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Empir. Methods Natural Lang. Process. Proc.*, Jun. 2014, pp. 1724–1734, doi: 10.3115/v1/d14-1179.

[29] Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," CoRR, vol. abs/1504.00941, pp. 1–9, 2015. [Online]. Available: http://arxiv.org/abs/1504.00941

[30] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[31] C. Scheuer *et al.*, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *Phys. Educ. Sport Child. Youth Spec. Needs Res. – Best Pract. – Situat.*, pp. 343–354, 2014, doi: 10.2/JQUERY.MIN.JS.

[32] B. Kedem, "Spectral analysis and discrimination by zero-crossings," *Proc. IEEE*, vol. 74, no. 11, pp. 1477–1493, Nov. 1986, doi: 10.1109/PROC.1986.13663.

[33] D. Enqing, L. Guizhong, Z. Yatong, and C. Yu, "Voice activity detection based on short-time energy and noise spectrum adaptation," in *Proc. Int. Conf. Signal Process.*, 2002, vol. 1, pp. 464–467, doi: 10.1109/ICOSP.2002.1181092.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, Sep. 2014, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1409.1556v6

[35] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. 2013 ACM Multimedia Conf.*, Spain, 2013, pp. 411–412, doi: 10.1145/2502081.2502245.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[37] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia Comput. Sci.*, vol. 112, pp. 2048–2056, Jan. 2017, doi: 10.1016/J.PROCS.2017.08.250.

[38] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors*, vol. 19, no. 7, Apr. 2019, Art no. 1733, doi: 10.3390/S19071733.

[39] M. Esmaeilpour, P. Cardinal, and A. L. Koerich, "Unsupervised feature learning for environmental sound classification using weighted cycle-consistent generative adversarial network," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art no. 105912, doi: 10.1016/J.ASOC.2019.105912.