CoVal-SGAN: A Complex-Valued Spectral GAN architecture for the effective audio data augmentation in construction sites

Michele Scarpiniti, Cristiano Mauri, Danilo Comminiello, Aurelio Uncini
Department of Information Engineering, Electronics and Telecommunications (DIET)
Sapienza University of Rome
via Eudossiana 18, 00185 Rome, Italy
{michele.scarpiniti, danilo.comminiello, aurelio.uncini}@uniroma1.it

Yong-Cheol Lee
Department of Construction Management
Louisiana State University
Baton Rouge, USA
yclee@lsu.edu

***Abstract*—Generative audio data augmentation in a construction site is a challenging area of research due to the high dissimilarity between work sounds of involved machines and equipment. However, it becomes necessary since the availability of audio data of critical work classes is often rare. Motivated by these considerations, in this paper, we propose a complex-valued GAN architecture working with the audio spectrogram, named CoVal-SGAN, for an effective audio data augmentation. Specifically, the proposed CoVal-SGAN exploits both the magnitude and phase information to improve the quality of the artificially generated audio signals and increase the overall performance of the underlying classifier. Numerical results, performed on data recorded in real-world construction sites, along with comparisons with other state-of-the-art approaches, show the effectiveness of the proposed idea by obtaining an improved accuracy.**

***Index Terms*—Audio data augmentation, Construction sites, Complex-valued architectures, Generative Adversarial Networks, GAN.**

I. Introduction

Construction Site Monitoring (CSM) is a challenging task for diverse project managers and individual workers, particularly safety managers. In this regard, an automated work and site supervision may provide great advantages to precisely organize the entire work flow and supervise equipment operations by calculating the exact amount of needed time, detecting the unnecessary costs, identifying negative issues, and ensuring the safety of all the workers. On the other hand, it is also indispensable for workers to avoid any safety risk [1].

Obviously, the monitoring of a complex work environment such as a construction site, is a difficult job involving security analysis, performance measurement, and resource management. To accomplish this goal, Automatic CSM (ACSM) systems have been recently introduced [2]. These systems are based on the automatic and independent analysis of diverse data collected from different sources using Machine Learning (ML) approaches [3].

This work has been supported by the project: "End-to-End Learning for 3D Acoustic Scene Analysis (ELeSA)" funded by Sapienza University of Rome Bando Acquisizione di medie e grandi attrezzature scientifiche 2018.

The most useful approaches are usually based on sensor data, in order to use the sensors' position to detect the locations of equipment, workers, and/or issues [1]. Effective activity recognition systems for construction equipment should also be based on data recorded from inertial measurement units (IMUs) [4]. Although the cheapness on such sensors, recorded data are often too noisy due to huge amount of vibrations in equipment, and, as consequence, the outcome of activity recognition is not always reliable. Among all sensors, camera-based systems have been largely used in ACSM [5], due to the maturity of video-based monitoring techniques. These techniques aim at detecting a target by using image processing algorithms. However, video sensors are limited to detecting a possible issue if it is far from the line of sight. Moreover, these systems require a high installation cost and a need of huge computation demand and large storage memories, which are required to address the huge amount of generated data.

Recently, motivated by a low cost of sensors, diverse studies investigated the potentials of audio-based ACSM and showed that they provide a promising solution for advanced unmanned field monitoring [3]. Microphone-based systems are significantly effective if they are no in the line of sight and to generate a not so high amount of recorded data. In addition, if multiple microphones are employed, the system may be also able to localize the active equipment by using beamforming techniques [6]. Other previous studies have also found that this new approach can obtain satisfactory performance and reliability [7]–[11]. In addition, since CSM by means of audio data is a subset of the Environmental Sound Classification (ESC) field, there are several effective Deep Learning (DL) approaches in the literature [12], like airport surveillance [13] and guns detection [14].

Among ML approaches, those based on Deep Learning (DL) techniques outperformed the shallow ones (like, support vector machines [15] or random forests [16]). Specifically, the most performing DL methods were those based on Convolutional Neural Networks (CNNs) [12], [17], [18], Deep Recurrent Neural Networks (DRNNs) [10], and others employing more

specific approaches, like Deep Belief Networks (DBNs) [11] and similar. All these approaches work by extracting meaningful features from the recorded audio signals [9] and/or by using directly the spectrogram representation [19]. Despite the advantages of DL approaches, they require a large set of training data.

Unfortunately, audio data recorded in a construction site are somehow limited in terms of the amount of data required for DL. This issue is further worsen by the fact that the real-world recordings are often extremely noisy including background and not relevant sounds, due to the natural characteristics of a construction work including multiple activities. A solution to deal with this issue may be represented by the recent techniques of data augmentation [20], usually based on the use of Generative Adversarial Networks (GANs) [21]. GANs are architectures composed of a couple of neural networks competing each other in order to generate new data with the same statistics as the training set [21].

Despite the frequent implementation of data augmentation techniques in computer vision and speech recognition domains [22]–[24], no use in ACSM has been so far performed, except the work in [25], which is focused on the augmentation of time-series data of IMUs. Since the GAN-based image augmentation has demonstrated a great effectiveness on the generation of hyper-realistic images [26], our idea is to apply GAN-based augmentation to the spectrograms of the audio data recorded in construction sites. Regretfully, most of the DL applications of spectrogram data only exploit the magnitude information, neglecting the phase one [27]. However, the phase information is essential in audio processing [28] and the simultaneous exploitation of both the magnitude and phase of the spectrogram may considerably improve the quality of the artificially generated audio signals and increase the overall performance of the classifier used for the ACSM. Some encouraging complex-valued approaches have been applied in radar [29] and medical [30] scenarios.

Motivated by these considerations, the main contributions of this paper are:

1) improving the classification accuracy of ACSM using audio data recorded in real-world construction sites;
2) proposing a novel complex-valued spectral GAN architecture that exploits both the magnitude and phase information, named CoVal-SGAN; and,
3) performing comparisons with other state of the art approaches.

The rest of the paper is organized as follows. Section II introduces some suitable GAN architectures that can be used for audio data augmentation. Section III describes the proposed complex-valued approach. Section IV introduces the experimental setup, while Section V shows the obtained numerical results and their comparison with other state-of-the-art approaches, discussed in Section VI. Finally, Section VII concludes the paper and outlines some future works.

II. GAN FOR AUDIO

A generative model has the task of generating samples belonging to the distribution of a given set of data. Among all the available models, Generative Adversarial Networks (GANs) have shown particularly good results.

A. GAN: Generative Adversarial Network

Introduced in 2014 by Goodfellows et al. [21], GANs are based on the use of two (deep) competing neural networks. The first network is called the *generator* and has the task of generating sufficiently realistic samples. The latter are subsequently placed in input to the second network, called the *discriminator*, which has the task of comparing the samples obtained from the generator with the real data coming from the dataset. The output of the discriminator will return a probability indicating whether the input data is real or not. The training is performed in order that the generator is able to generate realistic enough data to fool the discriminator, while the latter has to to recognize, in the best possible way, the fake samples from the real ones coming from the dataset. This process can be summarized in a *minimax* game between the two networks. The learning of these networks is therefore unsupervised, i.e. without the use of a set of real data to control the estimation error. Both networks are usually implemented by deep neural architectures.

In the classic GAN model, the input data z to the generator is noisy and belong to a $p_z(z)$ distribution. The generator $G(z;\theta_g)$ is a differentiable function represented by a deep network with parameters θ_g. The discriminator $D(x;\theta_d)$ is also a deep network that returns a scalar: $D(x)$ represents the probability that x comes from the dataset rather than $p_z(z)$. The work done by the generator in generating samples that are as realistic as possible is equivalent to learning the distribution of the data in the dataset, that is, making sure that the latter is equivalent to the distribution of the generator. In other words, it aims to minimize the likelihood that the discriminator will recognize a false sample generated: $\log(1 - D(G(z)))$. The discriminator can be seen as a classifier that must maximize the probability of assigning the false class to the samples from the generator and the true class to the samples from the dataset. Therefore the total cost function on which to apply the *minimax* game is the following [21]:

$$\begin{aligned}\min_G \max_D \mathcal{L}(D,G) = {} & E_{x\sim p_{data}(x)}\left\{\log D(x)\right\} \\ & + E_{z\sim p_z(z)}\left\{\log(1 - D(G(z)))\right\}.\end{aligned} \tag{1}$$

Although GANs work well in many situations, they show some limitations, like the vanishing gradient problem and the mode collapse.

In order to avoid these problems, in this paper we have chosen to use the loss function proposed in [31] for the Least Squares GANs (LSGANs). The cost functions that train the LSGANs are the following:

$$\min_D \mathcal{L}(D) = \frac{1}{2} E_{x\sim p_{data}(x)} \left\{ (D(x)-1)^2 \right\} + \frac{1}{2} E_{z\sim p_z(z)} \left\{ (D(G(z)))^2 \right\}, \quad (2)$$
$$\min_G \mathcal{L}(G) = \frac{1}{2} E_{z\sim p_z(z)} \left\{ (D(G(z))-1)^2 \right\}.$$

B. Suitable GAN architectures for audio

When used to generate synthetic audio samples, GANs usually work in the time-frequency domain. Specifically, the spectrogram representation is used as their input. Often, the spectrogram is treated as an image and the sole information on the magnitude is exploited, while the phase is ignored.

In literature, different GANs are suitable for the augmentation of audio samples. The most significant are summarized in the following. These approaches are also used as comparison against our proposed method (see Section V).

1) DCGAN: Convolutional Neural Networks (CNNs) are widely used in deep learning. When used as the generator and the discriminator of a GAN, we obtain a Deep Convolutional GAN (DCGAN) [32]. Authors of [32] showed that DCGAN are effective if the involved CNNs use only convolutional layers, without the pooling ones, the generator does not employ dense layers, and the batch-normalization is always used. DCGAN, exploiting spectrogram inputs, have been successfully used for audio synthesis [22], [33] and are known as SpecGAN. DCGAN (and hence SpecGAN) is trained by minimizing the LS loss in (2).

2) WCCGAN: GANs are not always able to properly extract meaningful features from a dataset. Sometimes, it is necessary that the generated samples have some inter-class or intra-class correlation with the original dataset. To this end, in the field of images generation, Zhu et al. [34] proposed to use a pair of GANs that work together in a cyclical fashion. This network has been called Cycle-Consistent GAN (CCGAN) [34]. In the work of Esmailpour et al. [23] an alternative model of CCGAN is proposed, which has been specifically designed for ESC and has been called Weighted Cycle-Consistent GAN (WCCGAN). Unlike the CCGAN, the architecture of a WCCGAN uses: i) the LS loss function in (2); and ii) a weighted bypass module for each generator. The latter, due to the LSGAN approach, are necessary in order to generate not excessively smoothed spectrograms, which can consequently be considered incorrect by the discriminator.

3) MelGAN-VC: A family of GANs used to generate audio samples is the MelGAN, which produces a synthetic audio signal by inverting in the time-domain the spectrogram generated in the mel scale [35]. These networks have a very different structure from the classic GANs. First of all, the generator input is not noisy but comes directly from a realistic distribution, i.e., the distribution of the mel spectrograms of the class to be generated. In addition, three discriminators are adopted. The architecture is equivalent for each of them, what varies is the type of input: the first discriminator works directly on the waveform generated while the other two work down-sampled versions. In this way, each discriminator learns features at different frequency ranges. Originally proposed for voice conversion, the variant MelGAN-VC [24] is capable of generating audio frames longer than the other GANs, since spectrograms are split in shorter segments and then concatenated before the discriminator. The quality of the generated audio is particularly good thanks to the use of Siamese networks [36]. This type of architecture is composed of two identical networks that force to preserve the distance between two data in the "feature vectors". In the particular case of the MelGAN-VC, Siamese networks are used at the input and output of the generator to preserve the spectrogram features after the action of the generator; in this way the generator keeps the contents of the entire spectrogram without using cost functions that depend directly on the spectrogram itself. The loss function related to Siamese networks is called TraVeL (Transformation Vector Learning) loss, whose complete definition and explanation can be found in [37].

III. Proposed approach

Despite the evident quantity of information available from the mel spectrum, the use of the magnitude alone may loose other useful information (phase), which can be important in the signal reconstruction or intelligibility. In fact, a spectrum can be defined by the magnitude and the phase; it has been shown [28] that the latter can play a fundamental role in the definition of signal properties in various applications [29], [30]. In particular, in audio the phase allows to understand the semantics of speech, the acoustic properties of a musical instrument, and the spatial position of the source. In the following, therefore, we want to analyze the signal by using the complex spectrogram, so to preserve those properties of the phase that can play an important role in both the quality of the generated samples and their classification.

The proposed approach is based on a complex-valued GAN in order to take into account also the phase information contained in the (mel) spectrogram representation. We propose a Complex-Valued Spectral GAN, named CoVal-SGAN. For a general overview on complex-valued deep networks, please refer to [38].

A. Complex-valued CNNs

Although the complex-valued deep networks have been used in a limited number of applications [29], [30], recently Trabelsi et al. in [38] propose an extension to the complex domain of the main building blocks of a deep network (such as the convolutional layer, the activation functions, and the batch normalization).

Let $x = x_R + jx_I$ be a complex-valued input signal, where x_R, x_I are the real and imaginary parts, respectively, and $j = \sqrt{-1}$ is the imaginary unit. In the same manner, let $W = W_R + jW_I$ be a complex-valued matrix collecting the weights of a complex convolutional layer. The complex-valued convolution $y = W * x$ between W and x can be defined as:

$$\begin{bmatrix} y_R \\ y_I \end{bmatrix} = \begin{bmatrix} W_R & -W_I \\ W_I & W_R \end{bmatrix} * \begin{bmatrix} x_R \\ x_I \end{bmatrix}, \quad (3)$$

where y_R and y_I are the real and imaginary parts of the output, respectively.

Several versions of activation functions in the complex domain have been proposed in the literature [39]. Most of the time, the activation operation defined in $\mathbb{R}$ is performed in an equivalent way on the real and imaginary parts, even if they do not always satisfy the Cauchy-Riemann conditions. In this paper we use the complex-valued extension of the ReLU (CReLU) activation, defined as follows:

$$\text{CReLU}(x) = \text{ReLU}(x_R) + j\,\text{ReLU}(x_I), \tag{4}$$

where $\text{ReLU}(\alpha) = \max\{0, \alpha\}$. A similar extension can be done for the Leaky ReLU activation. For the final classification layer, we use the *softmax* function. This is fed by the module of the last layer output.

Batch normalization (BN) is a very useful operation in CNNs, in order to stabilize the model and accelerate its convergence. The extension of the BN to the complex domain is implemented as vector whitening [38] and hence applied to the whited version $\widetilde{x}$ of the vector x. Similarly to the real-valued case, the complex BN is implemented as the scale and shift operation:

$$BN(\widetilde{x}) = \gamma\widetilde{x} + \beta, \tag{5}$$

where γ is a 2×2 positive semi-definite matrix:

$$\begin{bmatrix} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ir} & \gamma_{ii} \end{bmatrix},$$

and $\beta = \beta_R + j\beta_I$ is a complex parameter. For reciprocity, we have that $\gamma_{ri} = \gamma_{ir}$. Hence, the complex BN has a total of five parameters to be trained (γ_{rr}, γ_{ri}, γ_{ii}, β_R, and β_I).

B. The proposed CoVal-SGAN

The proposed CoVal-SGAN is based on the structure of a DCGAN, but exploiting the complex-valued layers introduced in the previous subsection. Specifically, the generator uses as input a complex-valued normal noise, while the discriminator's input is the spectrogram of a signal obtained from its STFT (which has values in $\mathbb{C}$ since we consider both the magnitude and phase information). The main differences with a DCGAN architecture, besides using complex-valued data, are as follows:

1) differently from the case of image processing, the output values do not have to be normalized since they represents samples of the (complex) STFT. Hence, the output layer uses a linear activation;
2) the generator input is generated as a Gaussian noise with variance greater than the unit: this trick has considerably reduced the convergence time of the proposed network, favoring the generation of realistic samples and in the right dynamic in about 100 epochs.

The generator and discriminator structures remain equivalent to those of the DCGAN; except for the generator kernels that are slightly larger. A picture of the general architecture of the proposed CoVal-SGAN is shown in Figure 1.

The proposed CoVal-SGAN architecture is fed by the STFT of the recorded audio signals and it has been trained minimizing the LS cost functions in (2) by using the Adam algorithm with a learning rate $\eta = 10^{-4}$ (obtained by a linear grid search in the interval $[10^{-5}, 10^{-3}]$) and the default value of the other parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). A batch size $B = 128$ is used and a total of 200 epochs have been performed. In order to check the effectiveness of the augmentation, the generated STFT samples have been fed in a complex-valued version of the CNN classifier presented in [18], and the 99.5% of the augmented samples, generated by the proposed architecture, was recognized as belonging from the true corresponding class.

IV. Experimental setup

A. The utilized dataset

Audio data of equipment operations, consisting of diverse construction machines and equipment, has been manually collected in real construction sites. A Zoom H1 digital handy recorder has been used for data collection. All files have been recorded by using a sample rate of $F_S = 44{,}100$ Hz and different files in WAV format for each machine are available. The different files are referring to a single machine per class recorded in similar environmental conditions.

Among the 25 classes available in the dataset, we have selected the seven classes shown in Table I with the corresponding time duration and equipment type. These classes include five excavators, a concrete mixer, and a compactor. The choice of using similar classes (five excavators) is made based on the need to test the generation of new samples, which must be as realistic and specific as possible, but also sufficiently different in order that the classifier be able to discriminate between different models.

From Table I, we can note that the used classes present different lengths: the file of the IRCOM compactor class is the longer one, while the excavator JD50D is the shorter one. Since the dataset is imbalanced, we expect that the accuracy of the JD50D class be low and generally it will tend to decrease the overall accuracy of the classifier, as demonstrated in [18]. In the rest of this paper, we use such a class for the data augmentation. Specifically, we augment the JD50D class in order to obtain one minute of additional data.

TABLE I: Available recording time (in [min:sec]), equipment type, and number of instances for each considered class.

Class	Time	Type	# of instances
CAT320E	05:00	Excavator	19,987
Concrete Mixer	14:59	Concrete Mixer	35,797
Hitachi50U	04:59	Excavator	19,982
IRCOM	05:00	Compactor	19,990
JC3CX	12:59	Excavator	20,545
JD50D	03:12	Excavator	12,770
JD50G	09:30	Excavator	37,981

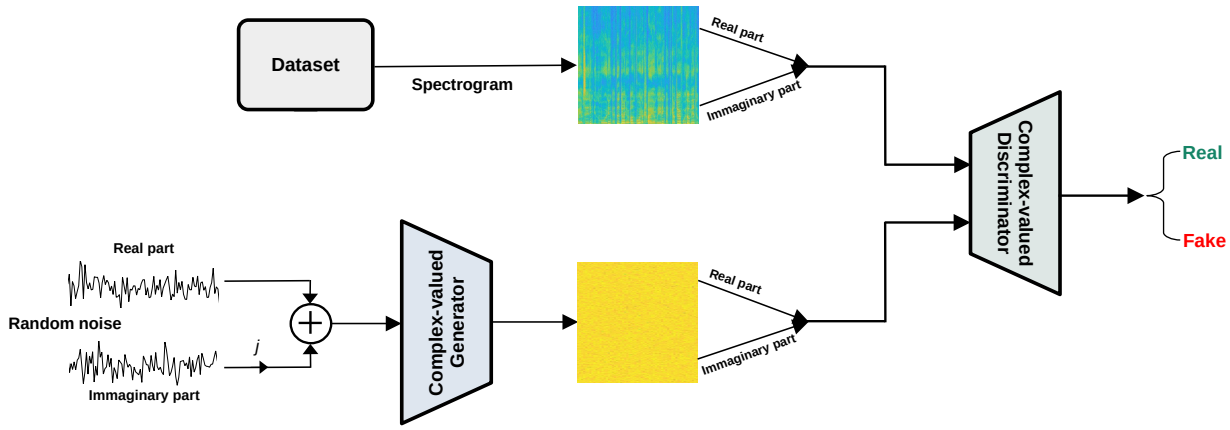


Fig. 1: General architecture of the proposed CoVal-SGAN.

B. Preprocessing

All files have been preemptively pre-processed and the silence segments (frames where the root mean square (RMS) is under the threshold of -30 dB) have been removed. The audio data has been resampled to 22,050 Hz and then the STFT and the (mel) spectrogram, with 128 bands, have been computed by using frames of 30 ms with 1024 samples, a hop size of 512 samples, and a number of FFT points equal to 1024. The number of instances extracted from the dataset is reported in the last column of Table I, for a total of 167,052.

In order to train, validate, and test our architecture, we split the original audio files into three parts: training samples (50% of the original dataset, corresponding to 85,132 instances), test samples (25% of the original dataset, corresponding to 40,960 instances), and validation samples (25% of the original dataset) used to find the optimal setting of the hyper-parameters.

Simulations have been carried out by using a computer equipped with an Intel® i7-8700K CPU @ 3.70GHz, with an NVIDIA GeForce GTX 1080. Software has been implemented in Python using Tensorflow for the models training and `librosa` library [40] for the waveform extraction and the spectrogram generation.

C. Baseline classifier and compared approaches

The baseline classification approach is based on the CNN classifier presented in [18]. The version implemented in this paper is composed of three convolutional layers, a dense layer, a dropout layer with a probability of 0.3, and a final softmax classifier. The input is the mel spectrogram of the recorded signal. This CNN classifier is trained by using the Adam algorithm, with a learning rate $\eta = 5 \times 10^{-5}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. A batch size $B = 128$ is used and a total of 500 epochs have been run.

Results are compared to those obtained by the SpecGAN [22], WCCGAN [23], and MelGAN-VC [24], discussed in Section II-B. In addition, a basic Pitch-Shift (PS) approach [20] has also been used to highlight the effectiveness of the GAN architectures. The pitch-shift consists in changing the pitch of an audio sample without varying its duration.

The performance of the proposed CoVal-SGAN[1] and the other approaches of the state of the art are evaluated in terms of the well-known metrics: accuracy, precision, recall, and F1-score.

V. Numerical results

In this section, we show the numerical results obtained by the CNN classifier using the proposed approach and the compared state-of-the-art ones for the augmentation of the JD50D class. All the considered approaches are used to generate 4000 additional spectrograms.

Table II shows the recall of each class for all the considered approaches. The second column is referring to the results obtained without any augmentation, i.e., applying directly the CNN classifier on the original datset.

From Table II, it can be seen how each approach has actually improved the recall on the augmented class, compared to the original dataset. Interestingly, CoVal-SGAN showed the best results, with a 3.7% increase in the recall of JD50D class compared to the non-augmented case. The least effective architecture was SpecGAN with only 1.1% increase in recall after the augmentation. Table II also highlights the effect that this augmentation had on the rest of the dataset. In fact, the increase in the classification accuracy on a class may consequently lead to worsen on the other non-augmented classes. Again, the CoVal-SGAN has the largest number of classes that have benefited from the augmentation on the JD50D. The worst results are related to the Pitch-Shift technique: although it increases the recall on the JD50D class, it also worsens the results on the rest of the dataset.

Table III shows the precision, recall, and F1-score obtained on the JD50D class by each compared approach. It emerges that the CoVal-SGAN obtains the best results for all the considered metrics, especially for the F1-score that increases of 1.8%. The Pitch-Shift technique is the one that presents the

[1]The CoVal-SGAN Python source code is available at: https://github.com/mscarpiniti/CoVal-SGAN

TABLE II: Results in terms of best recall [%] obtained on the considered classes for all the compared approaches.

Class	No aug.	Pitch shift	SpecGAN	WCCGAN	MelGAN-VC	CoVal-SGAN
CAT320E	86.0	83.5	85.6	86.0	85.5	86.2
Concrete Mixer	87.9	87.3	89.1	89.7	88.2	89.0
Hitachi50U	91.5	91.2	91.0	89.5	89.9	90.0
IRCOM	85.0	84.6	84.9	85.7	87.2	85.2
JC3CX	82.5	82.2	85.7	84.5	84.2	85.4
JD50D	79.4	81.7	80.5	81.1	82.4	83.1
JD50G	91.1	91.5	91.4	90.4	90.8	89.6

TABLE III: Precision, recall, and F1-score [%] on the only JD50D class for all the compared approaches.

	No aug.	Pitch shift	SpecGAN	WCCGAN	MelGAN-VC	CoVal-SGAN
Precision	86.3	82.0	85.5	85.6	85.5	85.9
Recall	79.4	81.7	80.5	81.1	82.4	83.1
F1-score	82.7	81.8	82.9	83.3	83.9	84.5

TABLE IV: Average accuracy [%] using the 5-fold cross-validation obtained by all the compared approaches.

	No aug.	Pitch shift	SpecGAN	WCCGAN	MelGAN-VC	CoVal-SGAN
Accuracy	86.45 $\pm$ 0.53	86.61 $\pm$ 0.63	86.78 $\pm$ 0.36	87.08 $\pm$ 0.29	86.96 $\pm$ 0.33	87.56 $\pm$ 0.28

TABLE V: Computational complexity in terms of number of millions of parameters and training time of the considered approaches.

Model	Parameters	Training time
SpecGAN	1.45	00:45
WCCGAN	4.20	02:40
MelGAN-VC	8.83	03:20
CoVal-SGAN	2.42	01:15

worst results: the precision is lowered by 4.3% and the F1-score decreases by about 1%, confirming what is observed in Table II.

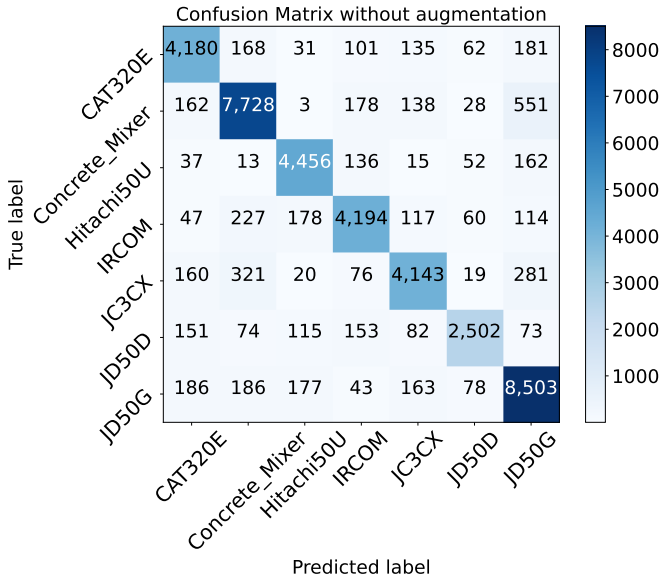
Table IV shows, for each approach, the average accuracy and standard deviation obtained using a 5-fold cross-validation. Each considered augmentation architecture is able to improve the average accuracy with respect the original dataset. However, the CoVal-SGAN, which provides the best mean accuracy, lowers the standard deviation by 0.25%, reflecting a greater uniformity and stability of the convergence of the classifier during the training phase. Also in this case, the worst results are obtained from the Pitch-Shift which increases the precision only by 0.16% but also increases the standard deviation by 0.1%.

Finally, Table V summarizes the computational time required for training the GANs aimed at generating the samples used for the augmentation, with the SpecGAN having the lowest time (i.e., about 45 minutes), while the MelGAN-VC is the slowest to train, with 3 hours and 20 minutes.

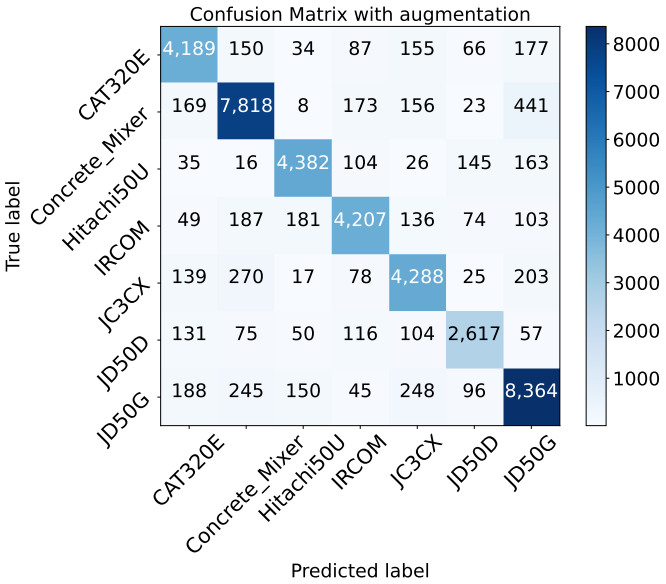
VI. Discussion

Each of the considered generative approaches helped to increase recall and accuracy across the dataset, proving a seemingly realistic sample generation. Although all the four compared GANs work quite well, CoVal-SGAN performs the best, showing a moderate but uniform increase, and leading the entire classification system to have greater accuracy on each class. The proposed architecture provides a recall nearly 4% greater in the JD50D class than the original dataset. This is also highlighted by the precision, F1-score, and mean accuracy shown in Tables III and IV. This behavior can be justified by the generation of more reliable spectrograms in CoVal-SGAN, which takes into account the real and imaginary parts of the spectrogram and not only its magnitude. This type of network allows to emphasize the spectral features that are lost if the spectrograms are generated through the observation of their magnitude only. Therefore, this result would seem to reaffirm the importance of the phase in audio applications, in particular in the generation of synthetic samples in the mel spectrogram domain. However, a careful observation of Table III shows that precision does not increase consequently, pointing out that some classes could be more confused after adding the new samples. This behavior is common to all the compared approaches, which exhibit a similar precision. From Table II, it can be seen that this phenomenon is mainly due to the Hitachi50U and JD50G classes. In fact, these classes show a reduced recall after the augmentation, proving that some of their instances are incorrectly classified as the JD50D class. The phenomenon can be graphically observed in Figure 2, which clearly shows that the confusion between Hitachi50U and JD50G classes and the augmented JD50D class is slightly increased.

A justification for this negative effect lies in the initial dataset (Table I). The Hitachi50U class is the smallest after the JD50D, so the augmentation of the latter class has resulted in the Hitachi50U having the least number of instances and being particularly penalized during the training. This effect can also be observed to a limited extent in the other two smallest classes CAT320E and IRCOM, but in the latter the worsening was not as pronounced as with the Hitachi50U. The probable



(a)



(b)

Fig. 2: Confusion matrices of: (a) the baseline approach [18], and (b) the proposed CoVal-SGAN.

reason for the little or no worsening of the CAT320E and IRCOM classes lies in the same cause of the recall decreasing of the JD50G class, namely the machine model. In fact, CAT320E and IRCOM are very different machines from the JD50D excavator: the first is a large excavator and the second is a compactor. As a result, they produce sounds dissimilar to the JD50D, which is a small excavator more similar to the Hitachi50U and JD50G. For this reason, CAT320E and IRCOM are easily distinguishable from the JD50D even with a lower number of instances. In contrast, the Hitachi50U and JD50G excavators are both similar in structure and size to the JD50D. In particular, the JD50G is a next model of the JD50D itself and therefore easily to be confused. However, in this case, the large number of samples available for the JD50G class allows the classifier to not exhibit a generally excessive deterioration as in the case of the Hitachi50U class.

Another important consideration is about the comparison between GANs and the Pitch-Shift (PS) augmentation method. In this case, an evident disadvantage is that, despite a recall increased by 2.3% on the augmented class, PS obtains a general deterioration on the other classes. Furthermore, the PS method is the only one that worsens the F1-score on the augmented class and provides a reduced accuracy. Probably, this effect is due to the wider distribution generated by PS: as a result, the likelihood that this distribution overlaps with other classes in the dataset increases. Although, in this case, the Hitachi50U and JD50G classes do not show substantial deterioration as in the case of the GAN, the global performance is the worst among all the considered approaches.

From the previous considerations, it can be said that the architecture that generally presents the best results is the proposed CoVal-SGAN. It significantly increases performance on the JD50D class and generally provides the best F1-score and averaged accuracy. The slightly lower performance on the two classes Hitachi50U and JD50G can be stemmed with a further augmentation to increase the number of their instances, in order to allow the final classifier to better discriminate the two models of the JD50x series. The only drawback of CoVal-SGAN is that the computational time is not negligible compared to other methods, like the SpecGAN, but, however, it is smaller than the WCCGAN and MelGAN-VC (see Table V). Hence, CoVal-SGAN may represent a good strategy for the augmentation of audio data in construction sites.

VII. Conclusion

In this paper, we propose a novel complex-valued spectrogram GAN (CoVal-SGAN) for the generative audio data augmentation in construction sites, in order to tackle the limited accuracy due to the imbalancement of some classes. The problem is particularly evident when using real-world data, given the difficulty in obtaining high-quality recordings of a large number of different machines and equipment. The proposed architecture exploits both the magnitude and phase information of the input spectrogram, allowing the generation of more realistic audio samples that increase the accuracy of the baseline CNN-based classifier. Some numerical results, performed on recorded data on real-world construction sites, along with comparisons with other state-of-the-art approaches, show the effectiveness of the proposed idea.

References

[1] J. Park, K. Kim, and Y. K. Cho, "Framework of automated construction-safety monitoring using cloud-enabled BIM and BLE mobile tracking sensors," *Journal of Construction Engineering and Management*, vol. 43, no. 2, p. 05016019, 2017.

[2] M. Kopsida, I. Brilakis, and P. A. Vela, "A review of automated construction progress monitoring and inspection methods," in *Proceedings of the 32nd CIB W78 Conference on Construction*, Eindhoven, Netherlands, 2015, pp. 421–431.

[3] B. Sherafat, C. R. Ahn, R. Akhavian, A. H. Behzadan, M. Golparvar-Fard, H. Kim, Y.-C. Lee, A. Rashidi, and E. R. Azar, "Automated methods for activity recognition of construction workers and equipment: State-of-the-art review," *Journal of Construction Engineering and Management*, vol. 146, no. 6, pp. 03 120 002–1 – 03 120 002–19, June 2020.

[4] C. R. Ahn, S. Lee, and F. Peña-Mora, "Application of low-cost accelerometers for measuring the operational efficiency of a construction equipment fleet," *Journal of Computing in Civil Engineering*, vol. 29, p. 04014042, 2015.

[5] H. Kim, Y. Ham, W. Kim, S. Park, and H. Kim, "Vision-based nonintrusive context documentation for earthmoving productivity simulation," *Automation in Construction*, vol. 102, pp. 135–147, June 2019.

[6] S. Scardapane, M. Scarpiniti, M. Bucciarelli, F. Colone, M. V. Mansueto, and R. Parisi, "Microphone array based classification for security monitoring in unstructured environments," *AEÜ – International Journal of Electronics and Communications*, vol. 69, no. 11, pp. 1715–1723, November 2015.

[7] C. Cho, Y.-C. Lee, and T. Zhang, "Sound recognition techniques for multi-layered construction activities and events," in *ASCE International Workshop on Computing in Civil Engineering*, vol. 2017, 2017, pp. 326–334.

[8] R. V. Sharan and T. J. Moir, "An overview of applications and advancements in automatic sound recognition," *Neurocomputing*, vol. 200, pp. 22–34, August 2016.

[9] Y.-C. Lee, M. Scarpiniti, and A. Uncini, "Advanced sound identification classifiers using a grid search algorithm for accurate audio-based construction progress monitoring," *The ASCE Journal of Computing in Civil Engineering*, vol. 34, no. 5, pp. 1–11, September 2020, paper 04020030.

[10] M. Scarpiniti, D. Comminiello, A. Uncini, and Y.-C. Lee, "Deep recurrent neural networks for audio classification in construction sites," in *28th European Signal Processing Conference (EUSIPCO 2020)*, Amsterdam, The Netherlands, 2020, pp. 810–814.

[11] M. Scarpiniti, F. Colasante, S. Di Tanna, M. Ciancia, Y.-C. Lee, and A. Uncini, "Deep belief network based audio classification for construction sites monitoring," *Expert Systems with Applications*, vol. 177, no. 114839, pp. 1–14, September 2021.

[12] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP 2015)*, Boston, MA, USA, 17–20 September 2015, pp. 1–6.

[13] M. Aldeman, R. Bacchus, K. Chelliah, H. Patel, G. Raman, and D. Roberson, "Aircraft noise monitoring using multiple passive data streams," *Noise & Vibration Worldwide*, vol. 47, no. 3–4, pp. 35–45, 2016.

[14] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, London, UK, 2007, pp. 21–26.

[15] C.-F. Cheng, A. Rashidi, M. A. Davenport, and D. V. Anderson, "Activity analysis of construction equipment using audio signals and support vector machines," *Automation in Construction*, vol. 81, pp. 240–253, September 2017.

[16] L. Yang and F. Su, "Auditory context classification using random forests," in *Proceedings of 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, 2012, pp. 2349–2352.

[17] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-end environmental sound classification using a 1D convolutional neural network," *Expert Systems with Applications*, vol. 136, pp. 252–263, December 2019.

[18] A. Maccagno, A. Mastropietro, U. Mazziotta, M. Scarpiniti, Y.-C. Lee, and A. Uncini, "A CNN approach for audio classification in construction sites," in *Progresses in Artificial Intelligence and Neural Systems*, ser. Smart Innovation, Systems and Technologies, E. A., M. Faudez-Zanuy, F. C. Morabito, and E. Pasero, Eds. Singapore: Springer, 2021, vol. 184, pp. 371–381.

[19] S. Chu, S. Narayanan, and C. C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 6, pp. 1142–1158, August 2009.

[20] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*, 2014, pp. 2672–2680.

[22] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Seventh International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, USA, 2019, pp. 1–16.

[23] M. Esmaeilpour, P. Cardinal, and A. L. Koerich, "Unsupervised feature learning for environmental sound classification using weighted cycle-consistent generative adversarial network," *Applied Soft Computing*, vol. 86, p. 105912, January 2020.

[24] M. Pasini, "MelGAN-VC: Voice conversion and audio style transfer on arbitrarily long samples using spectrograms," *arXiv:1910.03713v2*, 2019. [Online]. Available: https://arxiv.org/abs/1910.03713

[25] K. M. Rashid and J. Louis, "Times-series data augmentation and deep learning for construction equipment activity recognition," *Advanced Engineering Informatics*, vol. 42, p. 100944, October 2019.

[26] T. Zhang, W.-H. Tian, T.-Y. Zheng, Z.-N. Li, X.-M. Du, and F. Li, "Realistic face image generation based on generative adversarial network," in *16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, Chengdu, China, 2019, pp. 303–306.

[27] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, New Orleans, LA, USA, 5–9 March 2017, pp. 2721–2725.

[28] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 5, no. 69, pp. 529–541, May 1981.

[29] B. Vasudeva, P. Deora, and P. M. Bhattacharya, S. andPradhan, "Co-VeGAN: Complex-valued generative adversarial network for compressive sensing MR image reconstruction," *arXiv:2002.10523v3*, 2020.

[30] E. Cole, J. Cheng, J. Pauly, and S. Vasanawala, "Analysis of deep complex-valued convolutional neural networks for MRI reconstruction and phase-focused applications," *Magnetic Resonance in Medicine*, vol. 86, no. 2, pp. 1093–1109, 2021.

[31] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017)*, Venice, Italy, 2017, pp. 2794–2802.

[32] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations (ICLR 2016)*, San Juan, Puerto Rico, 2016.

[33] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Seventh International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, USA, 2019, pp. 1–17.

[34] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017)*, Venice, Italy, 2017.

[35] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Cana, 2019, pp. 1–12.

[36] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *23rd International Conference on Pattern Recognition (ICPR 2016)*, Cancun, Mexico, 2016, pp. 378–383.

[37] M. Amodio and S. Krishnaswamy, "TraVeLGAN: Image-to-image translation by transformation vector learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, Long Beach, CA, USA, 2019, pp. 8975–8984.

[38] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *Sixth International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada, 2018, pp. 1–19.

[39] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini, "Generalized splitting functions for blind separation of complex signals," *Neurocomputing*, vol. 71, no. 10-12, pp. 2245–2270, June 2008.

[40] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference (SciPy 2015)*, vol. 8, 2015, pp. 18–24.