Received 26 November 2020; | Revised 26 November 2020; | Accepted 26 November 2020

DOI: xxx/xxxx

ORIGINAL ARTICLE

Balanced Semi-Supervised GAN in Structural Damage Assessment from Low-Data Imbalanced-Class Regime

Yuqing Gao[1,2] | Pengyuan Zhai[1,3] | Khalid M. Mosalam*[1,2,4]

[1]Department of Civil and Environmental Engineering, University of California, Berkeley, CA, USA

[2]Tsinghua-Berkeley Shenzhen Institute (TBSI), Tsinghua University, Shenzhen, China

[3]Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA, USA

[4]Pacific Earthquake Engineering Research Center, University of California, Berkeley, CA, USA

***Correspondence**
Khalid M. Mosalam, 723 Davis Hall, University of California, Berkeley, CA 94720-1710, USA. Email: mosalam@berkeley.edu

Summary

In recent years, deep learning with convolutional neural networks to assess structural damages has gained growing popularity in vision-based structural health monitoring (SHM). However, large-scale structural image data acquisition and annotation processes are often costly in SHM. Moreover, the collected datasets are usually highly imbalanced in practice, because images of damaged structures/components are far rarer than those of intact ones. Hence, both data deficiency and class-imbalance hinder the practical performance. Common mitigation strategies include transfer learning, over-sampling, and under-sampling, yet these ad-hoc methods only provide limited performance boost that varies case by case. In this work, we introduce one variant of Generative Adversarial Network (GAN), named the Balanced Semi-supervised GAN (BSS-GAN). It adopts the semi-supervised learning concept and applies the balanced-batch sampling in training to relieve the low-data and imbalanced-class issues. A series of computer experiments on concrete cracking and spalling classification were conducted under low-data imbalanced-class regime with limited computing power. The results show that the BSS-GAN is able to achieve better damage detection performance (in terms of recall and F-β score) than other conventional methods, indicating its state-of-the-art performance.

KEYWORDS:
Structural damage assessment, Semi-supervised learning, GAN, Low-data, Imbalanced-class

1 | BACKGROUND & MOTIVATIONS

Nowadays, Machine Learning (ML) and Deep Learning (DL) lead the fashion and have benefited researchers of many disciplines. Since 2017, there has been an increasing number of DL studies and applications in civil and structural engineering. (Cha, Choi, & Büyüköztürk, 2017; Gao & Mosalam, 2018; Mosalam, Muin, & Gao, 2019; Oh, Kim, Kim, Park, & Adeli, 2017; Rafiei & Adeli, 2017; Yeum, Dyke, & Ramirez, 2018). Particularly, for vision-based structural health monitoring (SHM), the convolutional neural network (CNN) is proven to be a promising approach with high practical potential (Cha et al., 2017; Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018; Deng, Lu, & Lee, 2020; Gao, Li, Mosalam, & Günay, 2018; Gao & Mosalam, 2018; Jiang & Zhang, 2019; Liang, 2019; Maeda, Sekimoto, & Seto, 2016; Yeum et al., 2018; A. Zhang et al., 2017; C. Zhang, Chang, & Jamshidi, 2020). However, the frameworks in past studies are subject to three real-world challenges: (1) insufficient data, (2) imbalanced classes, and (3) limited computing power.

Most of the current DL applications in vision-based SHM fall into the supervised learning category, e.g., image classification (Cha et al., 2017; Gao & Mosalam, 2018), damage localization (Cha et al., 2018; Xue & Li, 2018) and segmentation (Yang et al., 2018), which requires a substantial amount of labeled data to reach the desired performance level. Obtaining large-scale labeled structural image datasets is costly and

labor-intensive. Compared with popular computer vision (CV) benchmark datasets such as ImageNet (1.5 million images) (Deng et al., 2009), MNIST (70,000 images) (LeCun, Bottou, Bengio, & Haffner, 1998) and CIFAR-10 (70,000 images) (Krizhevsky, 2009), current vision-based SHM datasets are far smaller. Although many previous studies (Cha et al., 2017; Dorafshan, Thomas, & Maguire, 2018) heavily relied on cropping techniques in hopes of augmenting the datasets, the cropped images had poor variety of invariant features, as they were sourced from similar scenarios in limited numbers of raw images. Additionally, the high similarity between cropped images in the training and test sets poses a risk of data leakage, as the model will memorize features in the training set that are simply repeated in the test set, which may exaggerate the model performance in real-world applications. Herein, the term "insufficient" has two meanings, namely the lack of data quantity and the lack of feature variety.

There also exists an imbalanced-class issue, which stems from the very nature of SHM: structural damages (due to either natural deterioration or extreme events such as earthquakes) are low-frequency occurrences. In real-world SHM image collection processes, damage-related data (e.g., cracking and spalling) usually only make up a small portion, and thus the majority of the data belong to the undamaged state. This leads to imbalanced datasets and further causes the model to be biased in favor of the "undamaged" class. On top of the already-existing low-data issue, the small "damaged" data portion easily leads to model overfitting (especially for high-dimensional image data). The impact of imbalanced datasets on DL performance in SHM has not been thoroughly studied, as many previous studies avoided this issue by constructing and training on balanced datasets.

Besides these two major issues, computing power limitations create another daunting challenge for efficiently training classification models. DL has benefited from the advancement of high-performance Graphics Processing Unit (GPU), the lack of which however becomes a curse in real-world applications. For example, the limited payload/carry-on capacity of a given small Unmanned Aerial Vehicle (UAV) (Villa, Gonzalez, Miljievic, Ristovski, & Morawska, 2016) due to budget limits or external environmental factors forbids the deployment of high-performance GPUs and supporting modules. Thus, to be able to conduct real-time recognition/inference with limited hardware, the network architecture needs to be degraded to a shallow, lightweight design, e.g., MobileNet (Howard et al., 2017), which, however, will compromise the classification performance. Besides, it is also worth noting that to pursue a better recognition performance, the network architectures (e.g., number of layers and number of filters) in past studies were usually designed and tuned specifically for their respective datasets after many trials and iterations. Tuning model parameters is costly, and may not be generic in practice.

Transfer learning (TL), over-sampling by conventional data augmentation (DA) methods, and under-sampling are common strategies to address the low-data and imbalanced-class issues. In TL, by tuning parameters from a pre-trained model, the new model can adapt to the target domain relatively easily, where the parameters in the early layers inherit certain knowledge from basic features in the source domain, making the model less dependent on extensive amounts of data (Pan & Yang, 2009). However, TL requires a pre-trained and open-source model from a source dataset, e.g., VGGNet (Simonyan & Zisserman, 2014) and ResNet (K. He, Zhang, Ren, & Sun, 2016) trained from the ImageNet dataset. Pre-trained networks are sometimes inaccessible for customized network designs and are usually computationally expensive to tune, e.g., VGGNet contains a large amount of trainable parameters. Moreover, TL only aims to mitigate the data deficiency problem and may not be sufficient to address the imbalanced-class issue.

In over-sampling, the minority-class data are over-sampled to reduce the majority-to-minority ratio by randomly duplicating minority-class samples or performing certain transformations or pre-processing operations, e.g., translation, flip, scale, whitening, and adding noise. The over-sampled minority class data are then mixed with the majority-class data to build a relatively balanced dataset (H. He & Garcia, 2009). However, conventional DA can only generate highly-correlated data, which does not increase feature variety, and additional storage space is needed for the over-sampled data. For some cases, inappropriately settings of DA might even lead to performance drops (Gao & Mosalam, 2020).

In under-sampling, majority-class data are randomly dropped to reduce the majority-to-minority ratio, which forms a balanced dataset with smaller size H. He & Garcia (2009). However, under-sampling may lead to an untrainable manner of the DL model or cause performance degradation due to information loss by ruling out a large amount of data.

Besides conventional methods above, Generative Adversarial Network (GAN) is considered an alternative DA method, where GAN generates synthetic data to augment the existing dataset and potentially enhance the model's performance (Gao, Kong, & Mosalam, 2019; I. Goodfellow et al., 2014; Salimans et al., 2016). Compared to TL, common GANs have simpler architectures and fewer trainable parameters than pre-trained DL models from ImageNet, which makes GANs applicable to custom-designed lightweight networks and less demanding for computing power. Compared with conventional DA, GAN can generate new data unseen by the model, increasing data variety. In vision-based SHM, there exist a few early GAN studies (Gao & Mosalam, 2020; Maeda, Kashiyama, Sekimoto, Seto, & Omata, 2020), and very little attention was directed

towards the imbalanced-class issue. In addition, the proposed GAN-based pipeline in (Gao et al., 2019) for classification problems is computationally inefficient. According to the findings in relevant GAN-based classification studies (Madani, Moradi, Karargyris, & Syeda-Mahmood, 2018b; Salimans et al., 2016), the semi-supervised learning mechanism exploits the features of the unlabeled data more thoroughly, simultaneously increasing the model's data generation and classification capabilities. Therefore, this has motivated us to convert the GAN into a semi-supervised variant.

Exploring GAN in practical engineering is still an open-ended discussion, and investigating the GAN-based methods in vision-based SHM is the main focus of this study. The main contributions of this work are:

- To simulate the restrictions of (1) low-data and (2) imbalanced-class, an extremely biased dataset containing limited images of undamaged state (UD), crack (CR) and spalling (SP) was built. To simulate the restriction of (3) limited computational resources, all pipelines were built on top of a shallow and general CNN classifier.
- One of previously proposed GAN-based classification pipeline, namely synthetic data fine-tuning (SDF) (Gao et al., 2019), was revisited herein with heuristic reasoning.
- A novel GAN-based classification pipeline with semi-supervised learning and balanced-batch sampling technique, namely the balanced-batch semi-supervised GAN (BSS-GAN), was proposed.
- Comparative computer experiments were conducted among BSS-GAN, baseline CNN (BSL), BSL using over-sampling, BSL with under-sampling, and other GAN-based augmentation methods.
- This study demonstrates that BSS-GAN improves the damage detection performance in terms of recall and F-β score and outperforms the above-mentioned methods under low-data and imbalanced-class regime.

This paper is organized as follows. Section 2 provides a literature review related to this study. Section 3 introduces several GAN-based augmentation methods. Section 4 describes the experimental objectives, dataset, evaluation metrics, network configuration and setups. Section 5 presents the experimental results and analysis. Finally, Section 6 delivers the conclusions and extensions of this study.

2 | RELATED WORK

The concept of GAN was first introduced by I. Goodfellow et al. (2014), which is a generative model that generates new data from the learned distribution. Unlike conventional DL models, GAN consists of two networks, namely the *generator* and the *discriminator*, where the generator creates synthetic data and the discriminator classifies an input sample as "real" or "synthetic." GAN uses adversarial training, where each network aims to minimize the gain of the opposite side while maximizing its own. Ideally, both the generator and the discriminator converge to the Nash equilibrium (I. Goodfellow, 2016), where the discriminator gives equal predictive probabilities to real and synthesized samples. Until now, GAN has been applied to many computer vision (CV) tasks, e.g., fake image generation (Radford, Metz, & Chintala, 2015), image-to-image translation (Isola, Zhu, Zhou, & Efros, 2017), and medical imaging synthesis, reconstruction, and classification (Yi, Walia, & Babyn, 2019).

In recent years, researchers start to use GAN to improve the DL model performance under the constrictions of low-data and imbalanced classes. In the general CV field, Antoniou, Storkey, & Edwards (2017) proposed Data Augmentation GAN (DAGAN), which overcomes the target domain's class imbalance issue by first finding representations of the source domain that are meaningful to generate other related data and then augmenting the target domain by new samples generated from the learnt representations. The performance of DAGAN-trained classifier was compared with other basic ones on Omniglot (Lake, Salakhutdinov, & Tenenbaum, 2015), EMNIST (Cohen, Afshar, Tapson, & Van Schaik, 2017) and VGG-Face (Parkhi, Vedaldi, & Zisserman, 2015) datasets using low-data setting and was shown to have promising enhancements. Mariani, Scheidegger, Istrate, Bekas, & Malossi (2018) proposed Balancing GAN (BAGAN) as an augmentation tool to restore balance in imbalanced datasets such that it can learn useful features from the majority classes and uses these to generate images for the minority classes. BAGAN was shown superior to ordinary GAN in terms of generated minority-class image quality and variability with imbalanced datasets. X. Zhang, Wang, Liu, & Ling (2019) introduced Deep Adversarial Data Augmentation (DADA) and formulated the DA problem into a supervised class-conditional GAN by developing a $2K$ discriminator loss function (where K is number of total classes), and enforced the generation of class-specific images. By generating images that are discriminable among classes, the discriminator can learn consistent decision boundaries from both real and synthetic data, which in turn improves the classification performance. The experimental results endorsed the superior capability of DADA in enhancing the generalization ability of DL models in extremely low-data regime. However, image data used in these studies are from general-purpose and well-cleaned CV datasets, which may not reflect the real environmental factors in practical scenarios.

As for real-world applications, GAN-based DA methods were also investigated in the medical imaging community. Frid-Adar, Klang, Amitai, Goldberger, & Greenspan (2018) applied Deep Convolutional GAN (DCGAN) to generate synthetic medical images for three lesion classes (Cysts, Metastases, and Hemangiomas) and then augmented the real image dataset by the generated images. It achieved 7% improvement over conventional DA. Similarly, for cardiovascular abnormality detection, Madani, Moradi, Karargyris, & Syeda-Mahmood (2018a) first down-sampled chest X-Ray images to 128×128 pixels and then used a variant of GAN to generate synthetic images to enrich the raw dataset. With the mixed images, the DL classifier reached a roughly 2% improvement over the non-augmented baseline and 1% over conventional DA. Furthermore, Madani et al. (2018b) formatted the GAN in a semi-supervised learning manner and conducted comparative experiments to investigate the effectiveness of their model under different magnitudes of labeled data. Their results indicated the high efficiency of such GAN pipeline under the low-data regime.

Instead of detecting human health conditions from medical images, vision-based SHM detects the health conditions of the structures. However, there only exist limited studies of vision-based SHM using GAN. The first documented work was conducted by Gao et al. (2019) where they designed a specific DCGAN architecture for structural images and proposed a leaf-bootstrapping training method to improve the quality of the synthesized images. Furthermore, based on validation experiments under low-data regime and limited computational resources, it was found that simply mixing synthetic images with the real ones did not work well, and might even lead to worse performance. Therefore, a special union training pipeline, namely synthetic data fine-tuning (SDF), was proposed, where the DL classifier was pre-trained on generated synthetic images and then fine-tuned by real ones. Such training pipeline was able to enhance the classifier performance by nearly 7% over the baseline. Recently, Maeda et al. (2020) applied GAN on road damage detection. They combined a progressive growing GAN (PG-GAN) (Bowles et al., 2018) along with Poisson blending, and then artificially generated road damage images used as new training data to improve the accuracy of road pothole detection tasks. However, as early studies, the above-mentioned works are still preliminary. They are computationally inefficient, because both the GAN and the CNN classifier need to be trained, and large amounts of synthesized images are generated, consuming extra computation time and storage space. Therefore in this work, we are motivated to reformulate the GAN in a more efficient semi-supervised fashion. Additionally, we also considered the class imbalance issue which has not been thoroughly investigated in previous studies.

3 | GAN-BASED AUGMENTATION METHODS

3.1 | Basics of GAN

GAN consists of a minimax game between the generator and the discriminator. Let $x \in \mathbb{R}^d$ be a sample, then $x_r \sim p_{data}$ is a sample from the the real data distribution and $x_g \sim p_g$ is a generated sample from the GAN-learned, synthetic data distribution. The generator G with parameters θ_G is trained to synthesize samples that mimic the real sample distribution, p_{data}, by mapping the noise vector (latent variable), $z \sim p_z$, to a synthesized sample $x_g = G(z; \theta_G)$, $x_g \sim p_g$. The discriminator D with parameters θ_D takes in a sample $x \in \mathbb{R}^d$ (either real or synthesized) and outputs $D(x; \theta_D)$, which is the predictive probability that x comes from p_{data} rather than p_g.

During the training, G and D compete with each other according to:

$$\min_G \max_D E_{x\sim P_{data}(x)} \left[\log(D(x))\right] + E_{z\sim P_z(x)} \left[\log(1 - D(G(z)))\right] \tag{1}$$

In Equation (1), the first term is the negated cross-entropy between $p_{data}(x)$ and $D(x)$, whose value is positively associated with D's ability of correctly predicting real samples as from the real data distribution $p_{data}(x)$; the second term is the negated cross-entropy between $p_z(z)$ and $1 - D(G(z))$, where $1 - D(G(z))$ is D's predictive probability that a synthesized sample $x_g = G(z)$ is indeed considered as "synthetic," i.e., $x_g \sim p_g$. D aims to maximize its discriminative power characterized by both terms, while the generator G tries to undermine D's performance by synthesizing realistic samples to trick D (minimizing the second term).

Both D and G can be parametrized by deep neural networks or CNNs, and they are trained and optimized alternatively according to Equation (1) until reaching the optima or designated number of iterations.

3.2 | Synthetic data over-sampling

GAN can be used to generate new data, which is thought to be useful in enriching the dataset. Therefore, one straightforward way is to over-sample the minority-class data by GAN-generated data to reduce the majority-to-minority ratio. This is named the GAN-based synthetic data over-sampling (GAN-OS), Figure 1.

However, preliminary investigations have demonstrated that such aggregation may sometimes render worse performance (e.g., more severe over-fitting) in the test set (Gao et al., 2019; Jain, Manikonda, Hernandez, Sengupta, & Kambhampati, 2018). Several possible reasons are: relatively lower image quality compared to the real ones, inherent data biases, and possible distribution differences between the synthetic and

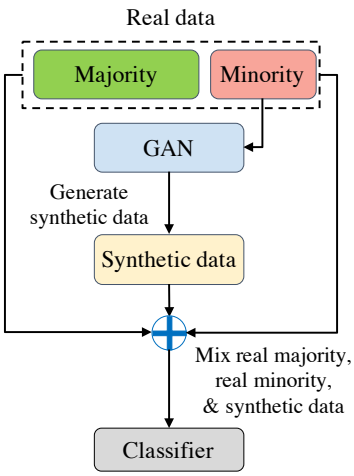


FIGURE 1 Illustration of GAN-based synthetic data oversampling (GAN-OS) pipeline

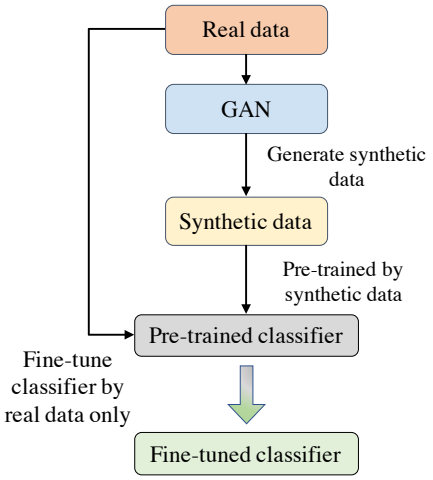


FIGURE 2 Illustration of SDF pipeline

the real ones. In our opinion, the dominating failure factor is the false sense of building a series pipeline.

The generated data are thought of as the fixed realization of the extra parameters introduced by the GAN model. From Figure 1, feeding a mixture of real and synthetic data into the classifier builds a series pipeline that passes the output of the GAN model followed by the real data into the classifier, which introduces additional parameters to the original classifier. Even though the generated data increase the number of data and possible variety, if the original classifier already suffers from over-fitting due to data deficiency, the additional parameters introduced to the series pipeline will increase the risk to further exacerbate the over-fitting, such as the case in (Gao et al., 2019).

3.3 | Synthetic data fine-tuning (SDF)

To alleviate over-parametrization in GAN-OS, Gao et al. (2019) proposed a pipeline based on TL, namely SDF. In SDF, a weak classifier is firstly pre-trained on the generated synthetic images and then fine-tuned by the real ones, Figure 2.

The intuition of the SDF pipeline is explained as follows. Using a weak or non-classic CNN classifier under computing power limitations usually implies that such a classifier does not have pre-trained parameters from ImageNet or other datasets in the source domain. Instead, its initialization fully depends on random, Gaussian, or other initialization approaches, e.g., Xavier initialization (Glorot & Bengio, 2010), which may be poor due to the large random parameter space. Conceptually, it is difficult to learn the decision boundary well by directly training from such initialization as illustrated in Figure 3. When considering the SDF pipeline, regardless of the correctness of the generated synthetic images, these images can be thought of as being generated from a similar or enlarged sample space, which origins from the real raw data. Therefore, it is believed that the classifier, when pre-trained from such enlarged space, can provide a better initialization for the subsequent fine-tuning step with the real data.

3.4 | Balanced-batch Semi-supervised GAN (BSS-GAN)

The original GAN is trained in an unsupervised learning manner, and its discriminator only differentiates unlabeled real samples from those synthesized by the generator. Herein, following the concept in (Salimans et al., 2016), a semi-supervised GAN can be formulated, where the output dimension (size) of the discriminator increases from 2 ("real" or "synthetic") to $K + 1$ such that the discriminator can classify samples from K real classes (for samples in $p_{data}(x, y)$) and one "synthetic" class (for generated samples in p_g). Besides labeled data from $p_{data}(x, y)$, the model can also utilize the unlabeled data from $p_{data}(x)$ simultaneously. These characteristics illustrate the core concept of semi-supervised learning.

3.4.1 | Discriminator loss

For each input sample x (either from p_{data}[1] or from p_g), the discriminator D outputs a $(K + 1)$-dimensional predictive probability vector $p_{model}(y|x)$, where $p_{model}(y = i|x) = \exp(D(x)_i) / \sum_{j=1}^{K+1} \exp(D(x)_j)$ represents the predictive probability of the i-th class and the $D(x)_1, \ldots, D(x)_{K+1}$ are logits output by $D(x)$ corresponding to each class. Herein, $p_{model}(y = K + 1|x)$ represents the predicted probability that sample x is "synthetic", and $D(x)$ in Equation 1 can be substituted by $1 -$

[1] P_{data} is a general form for both $P_{data}(x)$ and $P_{data}(x, y)$.

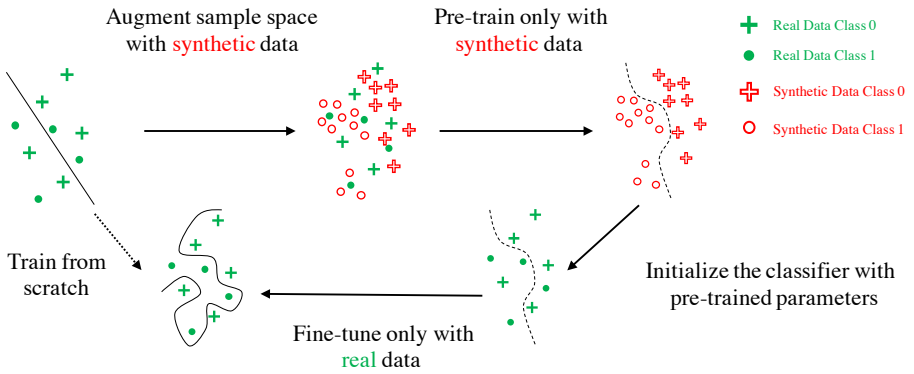


FIGURE 3 Two learning modes of conventional training and SDF

$p_{model}(y = K+1|x)$. Similarly, $1 - D(G(z))$ in the second term of Equation 1 is equivalent to $p_{model}(y = K+1|G(z))$. Then both terms are negated to form a minimization problem of D. The unsupervised loss without using any label information for $K+1$ classes is derived as:

$$L^{(D)}_{unsupervised} = -E_{x\sim P_{data}} \log(1 - p_{model}(y = K+1|x)) - E_{x\sim P_g} \log(p_{model}(y = K+1|x)) \quad (2)$$

It is noted that in Equation 2, since label information is not required, both labeled and unlabeled data can be used for the purpose of unsupervised feature learning.

For real and labeled data, the supervised discriminator loss is the cross-entropy between the real data-label distribution $p_{data}(x, y)$ and the model's predicted label distribution for K real classes (given real input sample x), $p_{model}(y|x, y < K+1)$:

$$L^{(D)}_{supervised} = -E_{x,y\sim P_{data}(x,y)} \log(p_{model}(y|x, y < K+1)) \quad (3)$$

Finally, the total discriminator loss is expressed as follows:

$$L^{(D)} = L^{(D)}_{unsupervised} + L^{(D)}_{supervised} \quad (4)$$

3.4.2 | Generator loss

The generator's objective is to "weaken" the discriminator's performance. However, the original formulation of the generator loss (I. Goodfellow et al., 2014) usually does not perform well. This is because the generator's gradient vanishes when the discriminator has high confidence of distinguishing generated samples from the real samples, i.e., when $D(G(z)) \to 0$. Therefore, the generator loss used in our formulation refers to the heuristic loss (I. Goodfellow, 2016), Equation 5.

$$L^{(G)}_{heuristic} = -\frac{1}{2} E_{z\sim p_z} log\, D(G(z)) \quad (5)$$

Instead of minimizing the expected log-probability of the discriminator being correct, the generator now maximizes the expected log-probability of the discriminator making a mistake, i.e., assigning a "real" label to a generated sample $G(z)$. In this study, the constant multiplier in Equation (5), i.e., $\frac{1}{2}$, is dropped and $E_{z\sim p_z} log D(G(z))$ is substituted by $E_{z\sim p_z} log[1 - p_{model}(y = K+1|G(z))]$ to accommodate the $(K+1)$-dimensional discriminator output, Equation. 6

$$L^{(G)}_{heuristic} = -E_{z\sim p_z} log[1 - p_{model}(y = K+1|G(z))] \quad (6)$$

Feature matching is a technique that prevents over-training the generator and increases the stability of the GAN (Salimans et al., 2016). It requires the generator to produce samples which result in similar features on an intermediate layer of the discriminator network as do the real samples. Therefore, the generator loss considering feature matching is formulated as:

$$L^{(G)}_{feature\ matching} = \left\| E_{x\sim p_{data}(x)} f(x) - E_{z\sim p_z(z)} f(G(z)) \right\|^2_2 \quad (7)$$

where $f(x)$ is the activation of an intermediate layer of the discriminator for a given sample x. In this study, $f(x)$ is defined by the ReLU (Nair & Hinton, 2010) activation on the flattened output of the last convolutional (Conv) layer of the discriminator network.

Finally, combining $L^{(G)}_{heuristic}$ and $L^{(G)}_{feature\ matching}$, the total generator loss is:

$$L^{(G)} = L^{(G)}_{heuristic} + L^{(G)}_{feature\ matching} \quad (8)$$

3.4.3 | Balanced-batch sampling

Class-imbalance issue not only affects the classification performance, but also deteriorates the perceptual quality and diversity of the generated samples (Mariani et al., 2018). During the conventional training (updating) procedure of a DL classifier or GAN, *mini-batch gradient descent* is commonly adopted (I. Goodfellow, Bengio, & Courville, 2016). Due to computational limitations, instead of using all the data at once, the DL network is only fed with one small batch containing m

data samples randomly selected from the full dataset of size N, where m is called the batch size and $m < N$. Statistically, if the dataset is imbalanced, the batch is also imbalanced, which eliminates neither the performance bias of the classifier nor the training instability of GAN.

Therefore, we introduce a small trick, namely *balanced-batch sampling* in training. As its name suggests, while forming the batch, the same amount of data is randomly sampled from each class. For balanced-batch sampling in GAN training, two types of balances are maintained in a given batch, namely (1) the balance among real classes in the labeled data, (2) the balance between any particular real class and the "synthetic" class. (1) means that n_l real labeled samples are randomly selected from K real classes, where $n_1^l, n_2^l, \dots, n_K^l$ are the numbers of data from each class, and $n_1^l = n_2^l = \cdots = n_K^l = n_l/K$. (2) means that the total amount of generated samples n_g matches any of the sub-batches from a certain real class, i.e., $n_g = n_l/K = n_k^l, k \in \{1, 2, ..., K\}$. In other words, each of the $K + 1$ classes contributes a sub-batch of the same size to the whole batch: $m = n_l + n_g = (K + 1) \cdot n_l/K$.

Moreover, the BSS-GAN can also utilize the unlabeled data in feature learning as by Equations 2 & 7. If additional unlabeled data are available, in any given batch, the ratio of unlabeled samples to any single-class sub-batch is controlled by a hyper-parameter c, i.e., $n_{ul} = c \cdot n_l/K$. In this study, to keep a smaller batch size to simulate computational limitation, $c = 1$ is used. Therefore, in each batch, $m = n_l + n_g + n_{ul} = (K + 2) \cdot n_l/K$.

3.4.4 | BSS-GAN algorithm

By formulating the GAN in a semi-supervised learning setting and using the balanced-batch sampling technique, the proposed integrated model is named the BSS-GAN. The BSS-GAN builds an end-to-end pipeline for both synthetic image generation and classifier training, and it is expected to have a stable and less biased performance under highly imbalanced datasets. Moreover, unlike training a supervised learning-based DL classifier, unlabeled data are also put into use. One example of using BSS-GAN in concrete crack detection is illustrated in Figure 4.

For a training batch size of m, the detailed training procedure of the BSS-GAN is as follows:

Step 0: Initialize the discriminator D and the generator G with θ_D and θ_G, respectively.

Step 1: A subset of the batch represented by real data (both labeled and unlabeled) is formed, $B_r = B_r^l \bigcup B_r^{ul} = \{(x_1^l, y_1^l), \dots, (x_{n_l}^l, y_{n_l}^l)\} \bigcup \{x_1^{ul}, \dots, x_{n_{ul}}^{ul}\}$, where n_l data-label pairs are equally sampled from the K real classes.

Step 2: Random noise vectors, $z = \{z_1, \dots, z_{n_g}\}$, are sampled from the noise prior $p_g(z)$. Then z is fed to G to generate n_l/K synthetic samples, $B_g = \{G(z_1), \dots, G(z_{n_g})\}$, which is the remaining subset of the batch.

Step 3: Feed B_r to D. For each $x_i \in B_r, i \in \{1, \dots, n_l + n_{ul}\}$, D outputs a K+1-dimensional probability vector, $u_i = [p_{model}(y = j|x_i), \forall j \in \{1, \dots, K + 1\}]^T$.

Step 4: Feed B_r^l to D. For each $(x_i^l, y_i^l) \in B_r^l, i \in \{1, \dots, n_l\}$, D outputs a $(K + 1)$-dimensional probability vector. However, only the first K dimensions are considered, $v_i = [p_{model}(y = j|x_i^l, j < K + 1), \forall j \in \{1, \dots, K\}]^T$.

Step 5: Feed B_g to D. For each $G(z_i) \in B_g, i \in \{1, \dots, n_g\}$, D outputs a K+1-dimensional probability vector $w_i = [p_{model}(y = j|G(z_i)), \forall j \in \{1, \dots, K + 1\}]^T$.

Step 6: Compute the discriminator loss, $L^{(D)}$:

$$\begin{aligned} L^{(D)} = &-\frac{1}{n_l + n_{ul}} \sum_{i=1}^{n_l+n_{ul}} log(1 - p_{model}(y = K + 1|x_i)) \\ &-\frac{1}{n_g} \sum_{i=1}^{n_g} log(p_{model}(y = K + 1|G(z_i))) \\ &-\frac{1}{n_l} \sum_{i=1}^{n_l} log(p_{model}(y = y_i^l|x_i^l)) \end{aligned} \tag{9}$$

Step 7: Compute the generator loss, $L^{(G)}$:

$$\begin{aligned} L^{(G)} = &-\frac{1}{n_g} \sum_{i=1}^{n_g} log[1 - p_{model}(y = K + 1|G(z_i)] \\ &+\left\| \frac{1}{n_l + n_{ul}} \sum_{i=1}^{n_l+n_{ul}} f(x_i) - \frac{1}{n_g} \sum_{i=1}^{n_g} f(G(z_i)) \right\|_2^2 \end{aligned} \tag{10}$$

Step 8: Optimize and update the network parameters θ_D and θ_G, where η is the learning rate.

$$\theta_D \leftarrow \theta_D - \eta \cdot \nabla_{\theta_D} L^{(D)} \tag{11}$$

$$\theta_G \leftarrow \theta_G - \eta \cdot \nabla_{\theta_G} L^{(G)} \tag{12}$$

Repeat steps (1) to (8) until convergence is achieved or the designated number of iterations is reached.

Balanced-batch sampling is only adopted in training. Once the BSS-GAN is trained, when referencing or predicting new data for classification purpose, all new data are fed into D only, and the predicted class is the one with the highest predictive probability among the first K entries of the D's output. Similarly, for synthetic data generation, the noise vector z is sampled and fed into G, which then outputs the synthetic data.

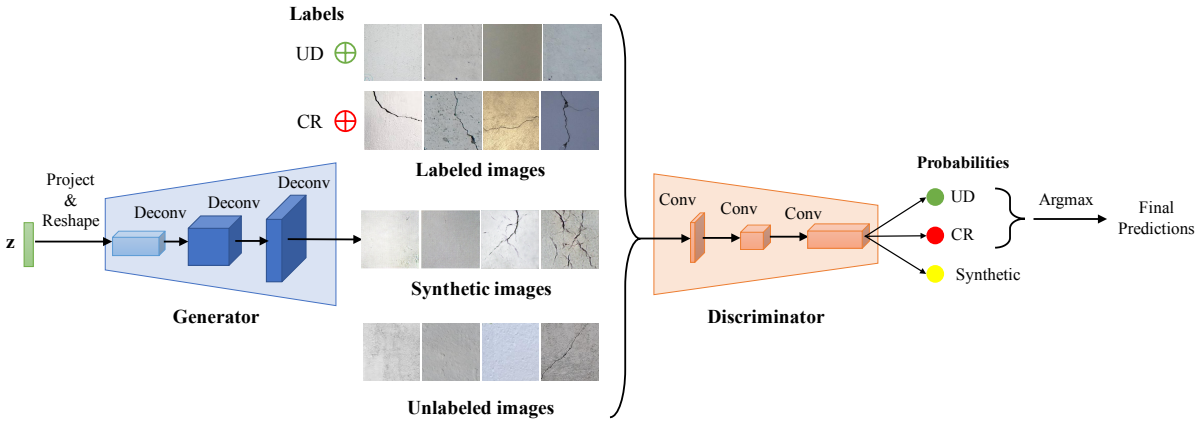


FIGURE 4 BSS-GAN pipeline in concrete crack detection

4 | EXPERIMENTAL PREPARATION

4.1 | Experimental objectives

In subsequent computer experiments, we aim to examine BSS-GAN's performance under low-data and imbalanced-class regimes on one common task in vision-based SHM, namely the concrete damage detection. Three key statuses, namely (1) undamaged state (UD), Figure 5a, (2) cracked (CR), Figure 5b, and (3) spalling (SP), Figure 5c, are considered, describing three damage levels in the order of increasing corrosion risk. In real-world applications, the class ratios of UD/CR and UD/SP are usually high. To simulate such imbalances in a realistic SHM data collection, an empirical ratio of 32:2:1 (UD:CR:SP) is selected for experimental purposes, where we also treat SP as less frequent than CR. Three major validation experiments are designed:

1. Binary crack detection with class a ratio of 16:1 between UD and CR.
2. Binary spalling detection with class a ratio of 32:1 between UD and SP.
3. Trinary damage pattern classification with a class ratio of 32:2:1 among UD, CR and SP.

Experiments (1) & (2) simulate the real-world binary damage detection in vision-based SHM, where the number of "undamaged" cases (UD) far exceeds that of "damaged" cases (CR or SP). Experiment (3) integrates the above two damage cases into a comprehensive but more complex three-class classification, which aims to evaluate the DL models in an imbalanced multi-class problem.

For a comparative study, in each experiment, multiple GAN-based pipelines are configured and compared:

1. A baseline shallow CNN classifier (BSL).
2. BSL with under-sampling the majority-class data to restore the class balance (BUS).
3. BSL with over-sampling minority-class data through conventional DA such as flips, translation and rotation (BOS-DA).
4. BSL with over-sampling the minority-class data by ordinary GAN-generated data (BOS-GAN).
5. BSL adopting the SDF training pipeline (BSL-SDF).
6. BSSGAN.

The performance of each case is evaluated by appropriate metrics, e.g., recall, confusion matrix, and F-β score, which are covered in more details in the following subsection. Beyond these, more intuitions are discussed in terms of (i) synthetic image quality of ordinary GAN and BSS-GAN, and (ii) effectiveness of unsupervised feature learning using different amounts of unlabeled data in BSS-GAN.

4.2 | Dataset

For generality, the images in this study were source from two open-sourced structural image datasets: PEER Hub ImageNet (ϕ-Net) (Gao & Mosalam, 2020) and SDNET2018 (Dorafshan et al., 2018). The structural images cover scenarios ranging from undamaged to extreme cracking or spalling. The images were further processed for the designed experiments:

1. Clean the dataset and select pixel-level (close up) images with high visual quality.
2. Select and store the images with label UD, CR, and SP to build the full dataset.
3. Rescale the images to 128×128 pixels with bicubic resampling.

(a) Undamaged **(b)** Cracked **(c)** Spalling

FIGURE 5 Sample images of three classes

Finally, a dataset with a total of 15,750 images was constructed including 14,400 UD, 900 CR, and 450 SP images with a class ratio of 32:2:1, Table 1. In addition, a 2:1 training-test split ratio was applied, so that enough test data (especially minority-class data) were assigned for a more proper evaluation. In experiment (1), UD and CR data were used, which has a 16:1 class ratio. For experiment (2), UD and SP data were used, which has a 32:1 class ratio. For experiment (3), all data were used. In all, compared with the ϕ-Net benchmark experiments (Gao & Mosalam, 2020) and general CV applications, the dataset used herein is considered imbalanced and low-data.

In addition, to investigate the contribution of unlabeled data in BSS-GAN, a smaller-scale imbalanced dataset for concrete crack detection was reformed, in which 20% of the labeled training data (2,040) in the original crack detection were used, and the remaining training data (8,160) were treated as unlabeled. Both labeled and unlabeled data still remained at the same 16:1 class ratio for UD:CR.

TABLE 1 Label statistics of the experimental dataset

Label	UD	CR	SP	Class ratio
Training	9,600	600	300	32:2:1
Test	4,800	300	150	32:2:1
Total	14,400	900	450	32:2:1

4.3 | Evaluation metrics

Classification accuracy (aka. overall accuracy) is defined by the number of correct predictions divided by the total number of predictions made for a dataset. However, it is not an informative measure for imbalanced classification problems. Because simply guessing all samples as from the majority class yields a misleadingly high accuracy. Thus, in this study, some more appropriate metrics are introduced and used.

4.3.1 | Confusion matrix

For classification, confusion matrix (CM) is a useful tool to summarize the model performance. For a binary classification problem, the CM has four possible outcomes: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). If further normalizing the CM entries by the number of predictions of each class, its diagonal entries become the true positive rate (TPR), Equation 13, and the true negative rate (TNR), Equation 14, which are used for evaluating the accuracy of detecting positive and negative classes respectively.

Additionally, recall, Equation 13, and precision, Equation 15, are also commonly used metrics. It is noted that in the binary case, recall is usually defined by the accuracy of correctly predicting the positive class, which is equivalent to TPR.

$$\text{TPR(Recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{14}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

Recall/TPR and precision measure different aspects of a model's performance. Recall is more appropriate if minimizing FN is the focus, and precision is more appropriate if minimizing FP is more important (Chawla, Japkowicz, & Kotcz, 2004). In damage detection, damaged state is usually defined as positive, and undamaged state is negative, and thus the number of negative data far exceeds the number of positive ones.

Failing to detect damages (more FN) bears more severe consequences than wrongly recognize undamaged samples as damaged (more FP). Thus, the first focus of the trained model should be to minimize the FN, measured by recall/TPR. On the other hand, precision is not an appropriate metric because due to the large number of negative (undamaged) data, a small drop in TNR will cause a large increase in FP, which overwhelms the TP and leads to a misleadingly low precision value. Furthermore, TNR can be an alternative to recall/TPR, which takes FP into account and measures the accuracy of correctly detecting negative class.

In addition, the normalized CM can also be applied to multi-class problems, i.e., $K > 2$, for which the recall for each class is evaluated and placed on the diagonal cells.

4.3.2 | F-β score

Besides CM, the F-β score is another suitable metric for imbalanced binary classification problems (Chawla et al., 2004). Instead of completely ignoring the precision, F-β weights and combines both precision and recall scores into a single measurement, Equation 16. Based on different β values, the β value measures different importance of recall over precision. When

$\beta = 1$, both recall and precision are weighted equally (F-1 score).

The β factor has some real-world interpretations (Chawla et al., 2004), e.g., how much higher the financial cost will be if failing to detect a damage than misproducing a false alarm. According to (Chawla et al., 2004), $\beta = 2$ is a common value ($F-2$ score). However, in SHM applications, the low tolerance of missing a damage calls for a much higher β. In this study, we consider $\beta = 2$ and $\beta = 5$, along with other β values in between.

$$\mathrm{F}_{\beta} = (1+\beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}} \tag{16}$$

4.4 | Network configurations

In many previous studies, the network design is sophisticated and task/dataset dependent. In order not to lose generality (while also keeping in mind the potential computing power limitations during field applications), BSL is built with a general multi-layer CNN discriminative classifier without a too elaborated network design and hyper-parameter tuning, refer to Table 2. For a fair comparison, the BSS-GAN's discriminator uses the same architecture as the BSL, e.g., same numbers of layers and filters, except for the output dimension ($K+1$ in BSS-GAN as opposed to K in the BSL). According to (Radford et al., 2015), Leaky ReLU (Maas, Hannun, & Ng, 2013) is used as the activation function with a negative slope coefficient $\alpha = 0.2$, and batch normalization (BatchNorm) (Ioffe & Szegedy, 2015) layers are inserted after the intermediate Conv layers with momentum 0.8. To avoid over-fitting, a 0.25 dropout rate is also applied.

The generator of the BSS-GAN is configured based on previous studies (Gao et al., 2019; Radford et al., 2015), Table 3. Due to the small image size (128×128), a conventional 100-dimensional noise vector is randomly generated from the Gaussian distribution as the input to the generator. There is no dropout in the generator, but BatchNorm layers with a 0.8 momentum are added after the deconvolutional (Deconv) layers except for the last one.

For the other two GAN-based pipelines (BOS-GAN and BSL-SDF), the GAN portions stay consistent with that of the BSS-GAN except for the loss functions, which are the loss functions used in the original GAN (I. Goodfellow et al., 2014).

4.5 | Experimental setups

In the first three experiments, all data were labeled. For the first five models, a batch size of 60 was used. For the BSS-GAN, the number of labeled real data was maintained at 60, for the remaining batch data, based on balanced-batch sampling, the numbers varied by cases. For binary cases, the total batch size of the BSS-GAN is $m = 60 + 60/2 = 90$. For three-class case, $m = 60 + 60/3 = 80$. In the last experiment investigating the usage of the unlabeled data, the batch size for binary crack detection is $m = 60 + 60/2 \times 2 = 120$.

All six pipelines were trained for 300 epochs, and saved for each epoch. In experiments 1, 2, and 4, the best model was selected by the highest TPR with over 90% TNR. For the three-class task of experiment 3, the best model was selected by the highest recall of SP with over 90% recall of UD. The optimizer was Adam (Kingma & Ba, 2014) with an initial learning rate of 2×10^{-5}. All experiments were conducted on the TensorFlow platform and performed on CyberpowerPC with single GPU (CPU: Intel Core i7-8700K@3.7GHz 6 Core, RAM:32GB & GPU: Nvidia Geforce RTX 2080Ti).

5 | EXPERIMENTAL RESULTS & ANALYSIS

5.1 | Experiment 1: Crack detection

From Table 4, the accuracy values for all six pipelines are higher than 90%, which are deceivingly promising. However, simply predicting all images as UD can easily lead to a 94.1 % overall accuracy under the 16:1 class ratio. More focuses should be placed on the TPR and TNR, which represent the accuracy of detecting CR and UD respectively. The resulting TPR and TNR are shown in Table 4 and Figure 6.

Starting from the low TPR of the BSL, it can be inferred that a shallow DL model can easily become biased due to extreme class imbalance (16:1). The BUS's under-sampling worked to some degree, as it improved the TPR from 31% to 46% without compromising the TNR too much. Similarly, BOS-DA's over-sampling helped increase the TPR to 45.7% without too much drop in the TNR, yet its TPR (along with that of the BUS) is unsatisfactory. BOS-GAN had the worst TPR (lower than the BSL), which conforms with the observations in (Gao et al., 2019) of the low performance of directly mixing synthetic data to the pipeline. BOS-GAN is extremely biased, and it is more prone to mis-predicting data as UD, causing a meaninglessly high TNR. Three factors lead to this poor and biased behavior: (1) the risk of introducing extra parameters mentioned in Section 3.2, (2) manually selecting augmented images is subjective, and (3) some GAN-generated images may be "adversarial" images (I. J. Goodfellow, Shlens, & Szegedy, 2014). For (3), although the GAN-generated images might be realistic-looking to human eyes, small feature perturbations undetectable by humans within these images might cause the classifier to make false predictions. Lastly, for the SDF pipeline, it obtained a similar performance to BUS and BOS-DA with a slight sacrifice in TNR to make up for the 3% enhancement in TPR. SDF is still biased, for which we can

TABLE 2 Configurations of the discriminator of the BSS-GAN

Layer	Filter size (#)	Activation	Shape	Notes (α: −ive slope coef. in Leaky ReLU)
Input	-	-	(N, 128, 128, 3)	Input RGB images of size 128×128
Conv	3×3 (32)	Leaky ReLU	(N, 64, 64, 32)	Stride = 2, $\alpha = 0.2$
Dropout	-	-	(N, 64, 64, 32)	Dropout rate = 0.25
Conv	3×3 (64)	Leaky ReLU	(N, 32, 32, 64)	Stride = 2, $\alpha = 0.2$
BatchNorm	-	-	(N, 32, 32, 64)	Momentum = 0.8
Dropout	-	-	(N, 32, 32, 64)	Dropout rate = 0.25
Conv	3×3 (64)	Leaky ReLU	(N, 32, 32, 64)	Stride = 1, $\alpha = 0.2$
Flatten	-	-	(N, 65,536)	65,536 = 32×32×64
Fc-layer	-	Softmax	(N, K)	BSS-GAN: $K = 2 + 1$; BSL: $K = 2$

TABLE 3 Configuration of the generator of the BSS-GAN

Layer	Filter size (#)	Activation	Shape	Notes
Input	-	-	(N, 100)	Noise generated from Normal distribution
Fc-layer	-	ReLU	(N, 131,072)	131,072 = 32×32×128
Reshape	-	-	(N, 32, 32, 128)	-
Deconv	3×3 (128)	ReLU	(N, 64, 64, 64)	Stride = 2
BatchNorm	-	-	(N, 64, 64, 64)	Momentum = 0.8
Deconv	3×3 (64)	ReLU	(N, 128, 128, 3)	Stride = 2
BatchNorm	-	-	(N, 128, 128, 3)	Momentum = 0.8
Deconv	3×3 (3)	tanh	(N, 128, 128, 3)	Stride = 1

infer that even though SDF improved model initialization, it did not help with solving the class imbalance issue.

In general, the five pipelines above are unsatisfactory in crack detection under the 16:1 (UD:CR) class ratio. All five pipelines have TPR below 50% and misleadingly high TNRs, implying severe biases towards the “undamaged" (UD) class. In other words, these pipelines can only detect less than 50% of all cracked structures or components, which is unacceptable in practice. On the contrary, the BSS-GAN model not only maintained an equally good TNR (over 90%) as others, but its TPR was also substantially higher (reaching 90%), indicating a nearly unbiased performance. Moreover, BSS-GAN is efficient in training, and it has an edge over other models in the following sense: compared with BUS, BSS-GAN can utilize all accessible data, which provide additional information; compared with BOS-DA, BOS-GAN, and SDF, the DA process is hidden and involved in the training process, with no extra data storage or multi-step training required.

F-2 & F-5 scores were then computed to take the recall (TPR) and precision into account. As aforementioned, the selection of β usually depends on its real-world interpretation. To avoid losing generality, F-β scores with varying β values are plotted in Figure 7. It is observed that starting from $\beta = 1$ (weighting recall and precision equally), BSS-GAN and SDF have increasing trends while BSL, BOS-DA, and BOS-GAN show decreasing trends. In addition, the $F - \beta$ values converge to recall scores (TPR) as β grows larger.

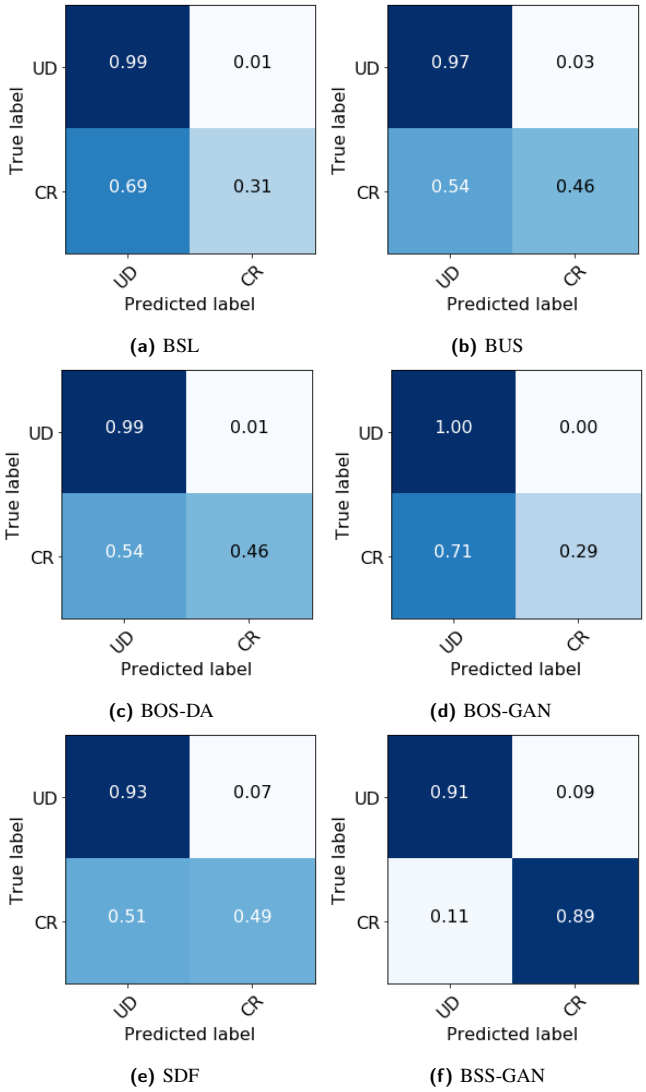
TABLE 4 Classification performance in crack detection (%)

Pipeline	TPR	TNR	F-2	F-5	Accuracy
BSL	31.0	99.4	35.2	31.7	95.4
BUS	46.0	97.0	46.6	46.1	94.0
BOS-DA	45.7	99.4	50.1	46.5	96.2
BOS-GAN	29.3	99.5	33.6	30.1	95.4
SDF	49.0	92.8	43.4	47.8	90.2
BSS-GAN	**89.3**	92.2	**72.8**	**85.6**	92.1

In SHM, it is more crucial to reduce FN than FP, so a large β is preferred. From Figure 7, as β becomes larger, BSS-GAN's $F - \beta$ exceeds those of other models with growing differences, suggesting its superiority in crack detection problems with high class imbalance.

5.2 | Experiment 2: Spalling detection

Even though the imbalanced class ratio (32:1 for UD:SP) in this experiment is twice higher than that of the crack detection task, from Table 5 and Figure 8, performance of all pipelines are better than the previous experiment. This observation can



(a) BSL **(b)** BUS **(c)** BOS-DA **(d)** BOS-GAN **(e)** SDF **(f)** BSS-GAN

FIGURE 6 Normalized CM in crack detection

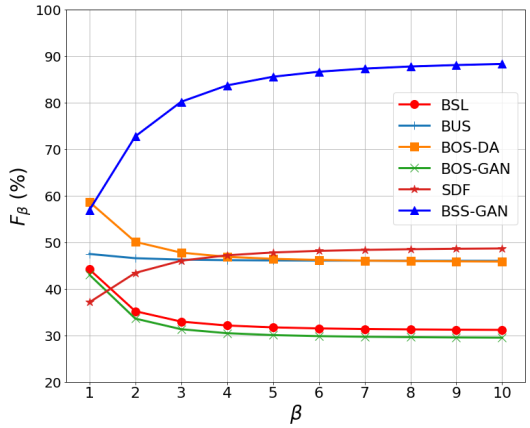


FIGURE 7 F_β with varying β in crack detection

TABLE 5 Classification performance in spalling detection (%)

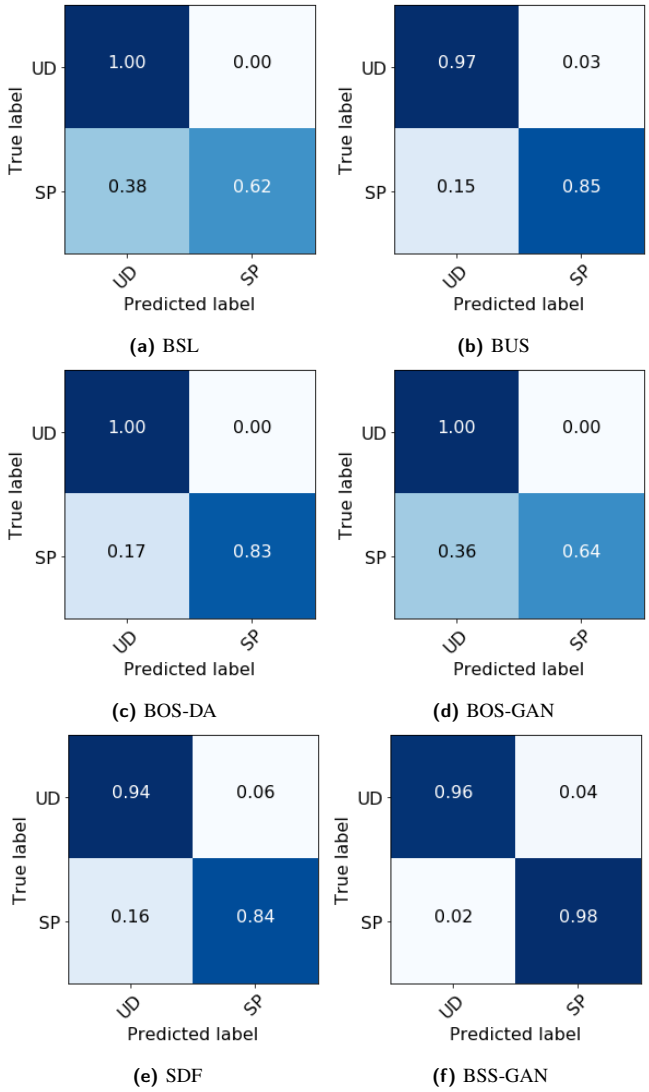
Pipeline	TPR	TNR	F-2	F-5	Accuracy
BSL	62.0	99.8	66.1	62.8	98.6
BUS	84.7	97.1	73.2	82.2	96.7
BOS-DA	83.3	99.9	85.5	83.7	99.4
BOS-GAN	64.0	99.9	68.6	64.8	99.0
SDF	84.0	92.8	62.3	78.7	93.8
BSS-GAN	**98.0**	95.8	77.6	**93.3**	95.9

be partially explained by different degrees of visual pattern similarity among UD, CR and SP. CR images (Figure 5b) are similar to UD images (Figure 5a) except for the appearance of surface fissures or fine cracks. On the contrary, SP images (Figure 5c) are more dissimilar, where the areas of spalling break the surface patterns in both color and texture, making the SP features more distinguishable. In this experiment, all pipelines obtained satisfactory TNR, and over 50% TPR values. As in the last experiment, BSL and BOS-GAN had the lowest TPR, which again showed the ineffectiveness of directly mixing synthetic data to the pipeline. BUS, BOS-DA, and SDF achieved similar performance, with TPR values reaching nearly 84%. BSS-GAN achieved the highest TPR (consistent with its crack detection performance). It not only maintained a high TNR (95.8%) as with other pipelines, but also improved the TPR from BSL's 62.0% to a surprising 98.0%, which is nearly 14% higher than those of BUS, BOS-DA, and SDF. The BSS-GAN outperformed the other five pipeline once more.

In spalling detection, due to high class imbalance, a small decrease in TNR will over-emphasize the increase of FP, leading to a higher F-2 score. For example, a mere 4.1% drop in TNR from BOS-DA to BSS-GAN makes BOS-DA have a higher F-2 score, although its TPR is 15% lower than that of BSS-GAN. Under the 32:1 (UD:SP) ratio, the F-2 score does not place enough emphasis on the recall (TPR), so a larger β, i.e., $\beta = 5$, is more informative and reasonable. According to Figure 9, both BSL and BOS-GAN share similar values and trends, while BUS, BOS-DA, and SDF have similar convergences after $\beta = 5$. Again, as β increases, especially beyond $\beta = 5$, BSS-GAN outperforms other pipeline significantly.

5.3 | Experiment 3: Damage pattern recognition

In this experiment, a more complex multi-class classification was investigated, where the imbalanced ratio for UD:CR:SP is 32:2:1. The performance results are listed in Table 6 and Figure 10. As with the two binary cases above, the BSL was biased in favor of UD with low recall values (around 30%) for both minority classes CR & SP. BOS-GAN was the second



(a) BSL (b) BUS (c) BOS-DA (d) BOS-GAN (e) SDF (f) BSS-GAN

FIGURE 8 Normalized CM in spalling detection

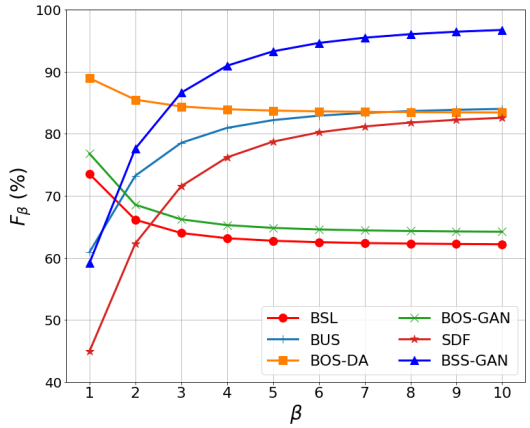


FIGURE 9 F_β with varying β in spalling detection

TABLE 6 Classification performance in damage pattern recognition (%)

Pipeline	Accuracy	Recall		
		UD	CR	SP
BSL	93.6	99.5	28.7	32.0
BUS	91.7	95.4	42.6	69.3
BOS-DA	95.2	99.5	29.0	90.0
BOS-GAN	88.3	92.7	31.3	60.7
SDF	89.6	93.5	35.0	73.3
BSS-GAN	89.7	90.9	**70.0**	**94.0**

worst pipeline in terms of class recall (31.3% and 60.7% for CR and SP respectively). BUS, BOS-DA and SDF did not perform well either, as their improvements in SP recall were merely sacrifices of the CR recall. For example, BOS-DA's SP recall reached 90%, but its CR recall remained low at 29%.

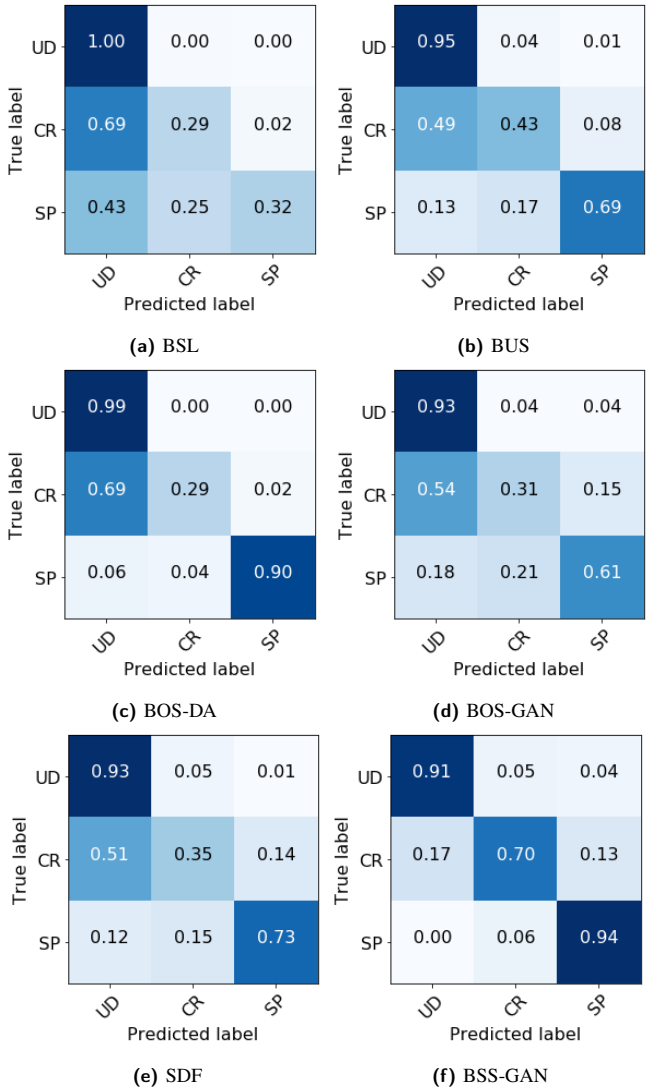
In general, the first five pipelines share a common drawback, that is the bias against CR, as characterized by their low CR recall values. This issue is attributed to the high visual similarity between UD and CR. GAN-based DA worsens this issue by generating images with blended features between UD and CR.

On the contrary, the BSS-GAN pipeline outperformed others with 90% UD recall, 70% CR recall, and 94% SP recall. Additionally, the BSS-GAN only misclassified 6% of the SP images as CR, and no SP images were misidentified as UD. It is thus much more reliable in detecting severer damages.

5.4 | Investigation on synthetic image quality

In this section, synthetic images generated from well-trained BSS-GAN models in the above experiments were compared with those generated by ordinary GAN in BOS-GAN and SDF pipelines, Figures 11, 12 & 13. It is noted that the generator in the BSS-GAN was trained using mixed-class images instead of class-specific (minority class) as in BOS-GAN and SDF, so it learned a mixed distribution of UD, CR & SP. For example, for crack detection, BSS-GAN was capable to generate both synthetic UD and CR images, Figure 11b.

Overall, there is no obvious mode collapse issue (the generator only produces limited varieties of images) in either the ordinary GAN or BSS-GAN. Besides basic visual features like textures and colors, the generator in both models can generate images with a variety of more complex features, e.g., crack orientation, location and width, and spalling shape, location, and area. As mentioned in (Gao et al., 2019), structural images have complex and mixed distributions, which makes it difficult for GAN to generate clear and class-discriminative images. However, conditioning operation (considering class information related to a specific class of images) makes the ordinary GAN capable of generating higher quality images towards that class.

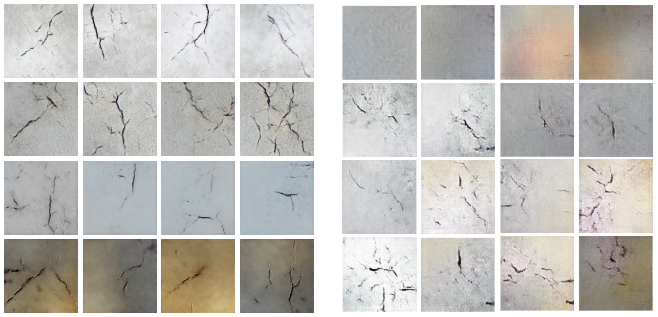


(a) BSL (b) BUS

(c) BOS-DA (d) BOS-GAN

(e) SDF (f) BSS-GAN

FIGURE 10 Normalized CM in damage pattern recognition



(a) Ordinary GAN (b) BSS-GAN

FIGURE 11 Sample synthetic images in crack detection

(a) Ordinary GAN (b) BSS-GAN

FIGURE 12 Sample synthetic images in spalling detection

For example, when training the GANs of BOS-GAN and SDF pipelines for crack detection, only feeding minority-class (CR) images can be viewed as one type of conditioning operation, which significantly reduces the data distribution complexity. Thus, in Figure 11a, the synthetic images show very realistic visual qualities. On the contrary, the generator in the BSS-GAN was trained with all images in an unsupervised manner. Thus, it had to learn a mixed distribution from both UD and CR. As a result, the synthetic images generated by BSS-GAN have features of UD, CR, or even the intermediate (mixed) state. To show this, in Figure 11b, synthetic images in the first row are smooth and resemble UD, while the remaining images resemble CR, but are somewhat blurry.

In addition, another possible explanation of the differences in image quality is the loss function. Unlike ordinary GAN, the loss function in BSS-GAN focuses more on classification than the quality of generated images. This is reflected by the supervised cross-entropy loss (Equation 3). On the contrary, the ordinary GAN only utilizes the unsupervised loss (Equation 2), which is more about feature learning than classification. These different training objectives influence the performance of the generator even though the same network architecture was used. It is thus inferred that BSS-GAN trades off its generator performance for more improvement in the discriminator's classification capabilities.

In summary, the ordinary GAN with class conditioning is able to generate higher-quality images than BSS-GAN by human judgement under 128×128 pixel resolution. However, according to the experimental results, the BSS-GAN was still able to learn meaningful representations from the generated images and improve its classifier performance. For BSS-GAN, the steps from synthetic image generation to feature learning and classification, are automatically and implicitly embedded in the training process, which is characterized by the game-theoretic competition between D & G. No human interaction is required (such as manual image selection as in BOS-GAN and SDF). These characteristics improve both training efficiency and the discriminative performance, which are essential enhancements for SHM decision making process under practical conditions of data deficiency and class-imbalance.

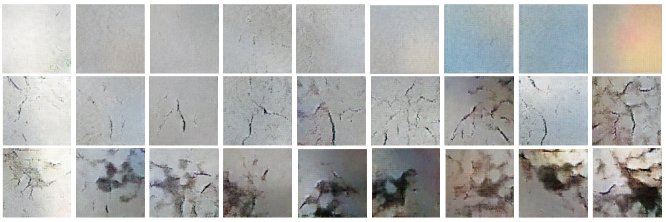


FIGURE 13 Sample synthetic images in damage pattern recognition generated by BSS-GAN

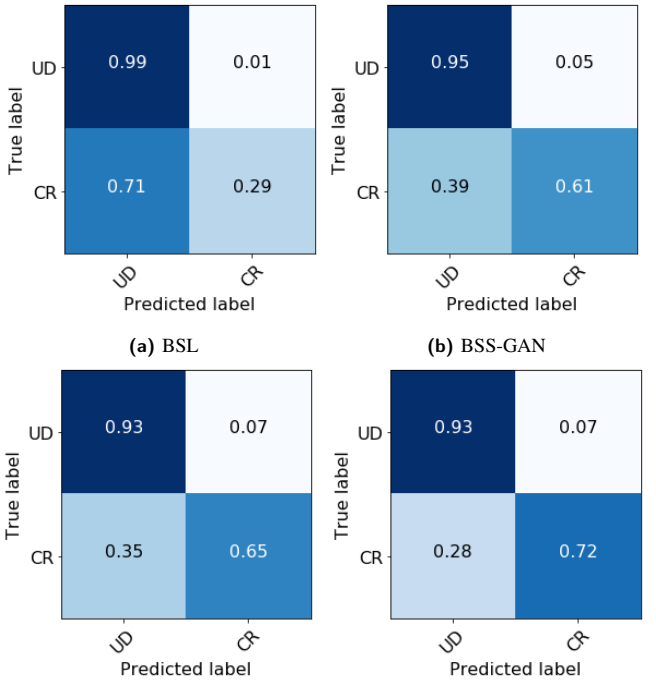
5.5 | Investigation on unsupervised feature learning

In this part, both BSL and BSS-GAN were initially trained using only 2040 labeled samples (20% of the total data). More unlabeled data were progressively added to subsequent BSS-GAN trials (50% and 100% of the remaining 8,160 samples). Results of the four cases are shown in Table 7 and Figure 14.

According to the results, initially given 20% of all data with imbalanced classes, BSL was biased with a TPR of only 29.0%. Although the TPR of BSS-GAN dropped to 60.7% compared to using full labeled dataset, it was still far less biased than BSL. BSL cannot improve beyond this point, as it can only learn from labeled data. However, as we introduced more unlabeled data to BSS-GAN (refer to Figures 14c & 14d), the TPR of BSS-GAN improved by 4% and 11% respectively by supplementing 50% and 100% of the unlabeled data. Although the supplementary data do not provide any label information, under the semi-supervised learning setting, BSS-GAN can still utilize information from the unlabeled samples. During the balanced batch sampling, the number of unlabeled data fed to each batch stays consistent with that of a single-class sub-batch, i.e., $n_{ul} = n_l/K$. As a result, even as more unlabeled data are introduced, they will never overwhelm the labeled samples during batch-by-batch training. Therefore, unlabeled data, once handled appropriately in each training batch, are able to supplement the learnt features and improve the classifier's performance, Figure 14d.

One seeming caveat is the decreasing TNR of BSS-GAN as more unlabeled data are supplemented. However, the upward trend of the F-5 score suggests that BSS-GAN models trained with more unlabeled data are better, which is based on our interest where high recall is prioritized over precision.

This experiment shows once more that the overall accuracy is a deceptive measure: the BSL achieved a 95.1% accuracy compared with BSS-GAN's 91.6% (with 8160 unlabeled samples), yet the BSL's performance is the worst overall in terms of TPR and F-5 score.



(a) BSL **(b)** BSS-GAN

(c) BSS-GAN using additional 50% unlabeled data **(d)** BSS-GAN using additional 100% unlabeled data

FIGURE 14 Normalized CM of four cases under a reduced-scale dataset with additional unlabeled data

TABLE 7 Classification performance in the study of unlabeled data utilization (%)

Pipeline	Unlabeled data	TPR	TNR	F-5	Accuracy
BSL	-	29.0	99.2	29.7	95.1
BSS-GAN	-	60.7	95.0	59.7	93.0
BSS-GAN	4,080	65.0	93.2	63.2	91.5
BSS-GAN	8,160	72.3	92.8	70.0	91.6

6 | CONCLUSIONS & EXTENSIONS

In this study, we firstly pointed out three key issues that impede the real-world applications of DL in vision-based structural damage assessment, namely data deficiency, class imbalance, and limited computing power. To address these issues, BSS-GAN, a semi-supervised learning GAN pipeline with balanced batch sampling, was proposed. It is an alternative to conventional or GAN-based data augmentation methods. To verify the effectiveness and efficiency of BSS-GAN in classification tasks, a series of computer experiments related to crack and spalling detection of reinforced concrete were designed and conducted under low-data and imbalanced-class regime. In addition, computing power limitations were simulated via using a shallow and generic CNN as the base design for all

pipelines. The experimental results were analyzed and compared with other five pipelines based on multiple metrics, i.e., recall (TPR) and F-β scores (F-2 & F-5). The synthetic image generation capabilities were then compared between BSS-GAN and the ordinary GAN. Lastly, the effectiveness of supplementing unlabeled data for feature learning in BSS-GAN was investigated.

The following key conclusions are drawn from the experiments:

- In general, BSS-GAN outperformed others in both binary crack and spalling detection under low-data and imbalanced-class settings. It achieved a significant improvement in TPR by reducing FN with only a slight decrease of TNR. BSS-GAN achieved better F-β scores, which put more weights on recall (TPR) over precision, e.g., F-5.

- Over-sampling the minority class by GAN-generated images (BOS-GAN) did not work well, and it led to worse performances than the baseline in spalling detection tasks. This was caused by three factors: (1) introduction of extra parameters, (2) subjective manual synthetic image selection, and (3) generation of "adversarial" images. These factors caused unstable training behaviors and exacerbated BOS-GAN's bias in favor of the majority class (UD). Such observations correlate to the findings in (Gao et al., 2019).

- BUS, BOS-DA, and SDF had similar but limited improvements over BSL, which were not satisfactory in practice. Their flaws include: in BUS, under-sampling eliminated a large portion of the labeled majority-class data, causing information loss; in BOS-DA, the conventional DA failed to increase feature varieties; in SDF, the model did not sufficiently address the imbalanced-class issue although it improved parameter initialization.

- In three-class classification, all pipelines except BSS-GAN were prone to predicting CR as UD (thus having low CR recalls). On the contrary, BSS-GAN obtained a promising CR recall of about 70%, while maintaining a high SP recall of 94% and a acceptable UD recall of 91%.Thus, it further indicates the stable and great potentials of the BSS-GAN in imbalanced multi-class tasks.

- BSS-GAN generated images of all classes without mode collapse, because it learned from a mixed-class distribution with balanced batch sampling. If only concerning generated image quality by human visual judgement, the generator in the ordinary GAN used in BOS-GAN and SDF was slightly better. It was inferred that the improvement of BSS-GAN's discriminator weakened its generator, but the generator was able to generate images realistic enough for the classifier to learn new features from.

- When labeled data have limited availability, the semi-supervised setting of the BSS-GAN allows it to utilize unlabeled data. With a proper ratio of unlabeled data placed into each training batch, BSS-GAN was able to capture meaningful information from the unlabeled data.

However, as the exploratory study, several aspects need to be investigated further:

- Even though pursuing a high precision is not so meaningful in extreme imbalanced-class problems, reducing FP is still desired, in order to lower the costs of false alarms. From experimental results, there exists much room of FP reduction for BSS-GAN.

- Our experiments have only shown that BSS-GAN is effective under the low-data regime. More experiments need to be conducted with medium-data and large-data regimes to explore its effective application scope.

- Only one general-purpose CNN architecture was tested in our experiments. More parametric studies with respect to network architectures are needed.

- Besides classification, damage localization and segmentation also face the issues of data deficiency and class imbalance. This suggests many other possible usage scenarios of BSS-GAN.

In conclusion, BSS-GAN is able to achieve state-of-the-art classification performance over conventional pipelines in all designed experiments under three major challenges: low data, imbalanced classes, and limited computing power. The promising results shed light on the great potential of semi-supervised GAN in vision-based damage assessment and SHM. This is clearly worth significant research efforts in the future.

ACKNOWLEDGEMENTS

The authors acknowledge the funding support of Tsinghua-Berkeley Shenzhen Institute (TBSI).

References

Antoniou, A., Storkey, A., & Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

Azimi, M., Eslamlou, A. D., & Pekcan, G. (2020). Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, *20*(10), 2778.

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., … Rueckert, D. (2018). Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*.

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 731–747.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, *6*(1), 1–6.

Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (ijcnn)* (pp. 2921–2926).

Deng, J., Lu, Y., & Lee, V. C.-S. (2020). Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 373–388.

Deng, J., Wei, D., Richard, S., Li-Jia, L., Kai, L., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 248–255).

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, *186*, 1031–1045.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using gan for improved liver lesion classification. In *Biomedical imaging (isbi 2018)* (pp. 289–293).

Gao, Y., Kong, B., & Mosalam, K. M. (2019). Deep leaf-bootstrapping generative adversarial network for structural image data augmentation. *Computer-Aided Civil and Infrastructure Engineering*, *34*(9), 755–773.

Gao, Y., Li, K., Mosalam, K., & Günay, S. (2018). Deep residual network with transfer learning for image-based structural damage recognition. In *Eleventh us national conference on earthquake engineering, integrating science, engineering & policy.*

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768.

Gao, Y., & Mosalam, K. M. (2020). Peer hub imagenet: A large-scale multiattribute benchmark data set of structural images. *Journal of Structural Engineering*, *146*(10), 04020198.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., … Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1125–1134).

Jain, N., Manikonda, L., Hernandez, A. O., Sengupta, S., & Kambhampati, S. (2018). Imagining an engineer: On gan-based data augmentation perpetuating biases. *arXiv preprint arXiv:1811.03751*.

Jiang, S., & Zhang, J. (2019). Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system. *Computer-Aided Civil and Infrastructure Engineering*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. *Technical Report TR-2009*.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 415–430.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, p. 3).

Madani, A., Moradi, M., Karargyris, A., & Syeda-Mahmood, T. (2018a). Chest x-ray generation and data augmentation for cardiovascular abnormality classification. In *Medical imaging 2018: Image processing* (Vol. 10574, p. 105741M).

Madani, A., Moradi, M., Karargyris, A., & Syeda-Mahmood, T. (2018b). Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In *2018 ieee 15th international symposium on biomedical imaging (isbi 2018)* (pp. 1038–1042).

Maeda, H., Kashiyama, T., Sekimoto, Y., Seto, T., & Omata, H. (2020). Generative adversarial network for road damage detection. *Computer-Aided Civil and Infrastructure Engineering*.

Maeda, H., Sekimoto, Y., & Seto, T. (2016). Lightweight road manager: smartphone-based automatic determination of road damage status by deep neural network. In *Proceedings of the 5th acm sigspatial international workshop on mobile geographic information systems* (pp. 37–45).

Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., & Malossi, C. (2018). Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. Cambridge: MIT press.

Mosalam, K., Muin, S., & Gao, Y. (2019). New directions in structural health monitoring. *NED University Journal of Research*, *2*, 77–112.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *In proceedings of the 27th international conference on machine learning* (pp. 807–814).

Oh, B. K., Kim, K. J., Kim, Y., Park, H. S., & Adeli, H. (2017). Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. *Applied Soft Computing*, *58*, 576–585.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2).

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234–2242).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Villa, T. F., Gonzalez, F., Miljievic, B., Ristovski, Z. D., & Morawska, L. (2016). An overview of small unmanned aerial vehicles for air quality measurements: Present applications and future prospectives. *Sensors*, *16*(7), 1072.

Xue, Y., & Li, Y. (2018). A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. *Computer-Aided Civil and Infrastructure Engineering*, *33*(8), 638–654.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1090–1109.

Yeum, C. M., Dyke, S. J., & Ramirez, J. (2018). Visual data classification in post-event building reconnaissance. *Engineering Structures*, *155*, 16–24.

Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical image analysis*, 101552.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., … Chen, C. (2017). Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zhang, C., Chang, C.-c., & Jamshidi, M. (2020). Concrete bridge surface damage detection using a single-stage detector. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 389–409.

Zhang, X., Wang, Z., Liu, D., & Ling, Q. (2019). Dada: Deep adversarial data augmentation for extremely low data regime classification. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 2807–2811).

How to cite this article: Gao Y., Zhang Y., and Mosalam K.M. Balanced Semi-Supervised GAN in Structural Damage Assessment from Low-Data Imbalanced-Class Regime. *Computer-aided Civil and Infrastructure Engineering*, 2020; 00:1–14.