Balanced Semi-supervised Generative Adversarial Network in Vision-based Structural Damage Assessment under Imbalanced-class and Low-data Regime

Y. Gao[1], K. Mosalam[2], P. Zhai[3]

[1] PhD Candidate, Department of Civil and Environmental Engineering, University of California, Berkeley, CA, United States & Tsinghua-Berkeley Shenzhen Institute (TBSI), Shenzhen, China, gaoyuqing@berkeley.edu

[2] Taisei Professor of Civil Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA, United States & Tsinghua-Berkeley Shenzhen Institute (TBSI), Shenzhen, China, Mosalam@berkeley.edu

[3] Undergraduate Student, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA, United States, alphabilly@berkeley.edu

Abstract

In recent years, using deep learning (DL) with convolutional neural network (CNN) to assess structural damages has gained growing interest in vision-based structural health monitoring (SHM). However, most of the current DL applications in SHM fall into the supervised learning category, e.g., image classification, damage localization, etc., which requires large amounts of labeled data. Obtaining such large-scale labeled dataset is costly and requires significant efforts. In practice, the collected structural image dataset is usually highly imbalanced, because the images related to damages, e.g., reinforced concrete cover spalling, are far fewer than those without structural damage patterns, which further exacerbates the training difficulty. Transfer learning (TL) and data augmentation (DA) are common strategies to mitigate the above-mentioned issues. However, the former requires fine-tuning a deep pre-trained network with large amounts of parameters, which not only is computationally expensive but also requires an open-source, pre-trained model; the latter can only generate highly-correlated data, which does not increase data variety. Generative Adversarial Network (GAN), a recent computer vision technology, is an alternative to TL and DA for increasing model performance, especially under deficient data and limited computational resources. In this work, we introduce one variant of GAN, named the Balanced Semi-supervised GAN (BSS-GAN), which adopts the semi-supervised learning concept and uses the balanced-batch sampling technique during training. It aims to better-exploit GAN-generated data to enhance both classification accuracy and computational efficiency. We design and conduct a series of computer experiments with respect to spalling detection under the imbalanced-class and low-data regime, i.e., training with only 1,130 labeled images with a roughly 4:1 class ratio for non-spalling : spalling. During the experiments, BSS-GAN is compared with a baseline CNN (BSL), which uses the conventional training procedure, under different magnitudes of labeled data. Moreover, BSS-GAN is compared with another GAN-based augmentation pipeline (GAN-Aug), which trains the BSL on a balanced dataset by oversampling the minority class with GAN-generated synthetic data. The results show that BSS-GAN achieves about 92% overall accuracy and about 84% true positive rate, indicating its improved classification performance over both BSL and GAN-Aug. This demonstrates the effectiveness and research potential of semi-supervised GAN for future vision-based SHM.

Keywords: Deep Learning, Generative Adversarial Network, Semi-supervised Learning, Structural Health Monitoring.

1. Introduction

As the concept of structural resilience becomes more popular, the needs for structural health monitoring (SHM) are no longer limited to detecting structural damage after natural disasters or extreme events, and SHM becomes more crucial for regular inspection and maintenance during the operation of a structure. Usually, SHM efforts heavily rely on the domain expertise of well-trained engineers, whose manual inspections on the structures is often inefficient. Therefore, the state-of-the-art artificial intelligence (AI) technologies, e.g., deep learning (DL), are gaining more attention in SHM research. There are two major directions in SHM: vibration-based and vision-based SHM. Particularly for vision-based SHM, computer experiments have achieved great success [1-4] in structural damage assessment, indicating high practical potentials.

However, two key issues impede the development of DL in vision-based SHM: insufficient data and imbalanced classes. Most of the current DL applications in vision-based SHM fall into the supervised learning category, e.g., image classification, damage localization, etc., which requires a substantial amount of labeled data. Obtaining large-scale labeled image datasets of structures is costly and time-consuming. Moreover, unlike common image benchmark datasets, e.g., MNIST [5], CIFAR-10 [6], SHM datasets have highly imbalanced classes. Such imbalance originates from the nature of SHM: structural damages due to either natural deterioration or extreme events such as earthquakes are less frequent cases due to design code objectives and high levels of conservativism for safety purposes. Past studies tend to ignore this issue by constructing relatively balanced datasets.

Transfer learning (TL) and data augmentation (DA) are conventional strategies to overcome data deficiency. 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 large amounts of data [7]. In DA, the number of training samples is increased via a series of affine transformations or other pre-processing operations, e.g., translation, flip, scale, whitening, and adding noise. Both TL and DA have significant curtailments: the former requires fine-tuning a deep pre-trained network with a large amount of parameters, which not only requires a pre-trained, open-source model but is also computationally expensive; the latter can only generate highly-correlated data, which does not increase data variety and sometimes might lead to worse results [8].

Generative Adversarial Network (GAN) is an alternative to TL and DA, but very few experiments have been conducted under low-data regimes and limited computational resources. GAN generates synthetic data from the real data distribution, augmenting the data and possibly enhancing the model performance. Compared to TL, common GAN designs have simpler architectures and fewer trainable parameters than pre-trained DL models from ImageNet [9], e.g., VGGNet [10] and ResNet [11], which makes GAN applicable to custom-designed networks and less demanding for computing power. Compared to DA, GAN can generate new data unseen by the model, increasing data variety. In vision-based SHM, there exist a few early GAN studies [12], which only explore very basic GAN usage with inefficient algorithms. In this work, we convert the GAN into a semi-supervised variant as suggested in [12], where the semi-supervised mechanism can more thoroughly exploit the features of the unlabeled data, simultaneously increasing the model's data generation and classification capabilities.

In this study, the authors aim to build a semi-supervised GAN for structural damage assessment and to evaluate its performance compared with the baseline DL classifier through multiple comparative experiments. To overcome the imbalanced-classes issue, a balanced batch sampling technique is adopted in the training procedure. We name this model the Balanced Semi-supervised GAN (BSS-GAN).

2. Related work

The concept of GAN was first introduced by Goodfellow et al. [13] in 2014, which is a generative model that generates new data from learned data 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". More details are covered in Section 3. 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, where the discriminator gives equal predictive probabilities to real and synthesized samples. The rapid growth of GAN studies has produced many types of GANs, which modify the ordinary GAN [13,14] with different network architectures, loss functions, training strategies, etc. Some notable ones are the Deep Convolutional GAN (DCGAN) [15], Wasserstein GAN (WGAN) [16], Conditional GAN [17], and BigGAN [18].

Until now, GAN has been applied to many computer vision (CV) tasks, e.g., fake image generation [15,18] and image-to-image translation [19]. It is noted that GANs have seen rapid adoption in the medical imaging [20] community for purposes of image synthesis, reconstruction, and classification. A few examples are reviewed here. Frid-Adar et al. [21] applied DCGAN to generate synthetic medical images for three lesion classes (Cysts, Metastases, and Hemangiomas) and then augmented the real image dataset with the generated images, which achieved 7% improvement over classic augmentation methods. Similarly, for cardiovascular abnormality detection, Madani et al. [22] 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 traditional DA. Furthermore, Madani et al. [23] formatted the GAN in a semi-supervised manner and conducted comparative experiments to investigate the effectiveness of GAN under different levels of labeled data. Their results indicated such GAN worked more efficiently under low-data regimes.

Instead of detecting human health conditions from medical images, vision-based SHM detects the health conditions of structures. However, there exist limited studies in the SHM using GANs. The first documented work was conducted by Gao et al. [12] in 2019 where they designed a specific DCGAN architecture for structural images and proposed a leaf-bootstrapping 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, which might even lead to worse performance. Therefore, a special union training pipeline 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. However, this training pipeline is inefficient, because both the GAN model and the classifier need to be trained, and large amounts of synthesized images are left valueless after the pre-training process. From the findings of Gao et al. [12] and the study of Madani et al. [23], reformulating the problem into a semi-supervised learning paradigm seemed to be a promising solution.

3. Methodology

3.1 Ordinary GAN

Goodfellow et al. [13] proposed a generative adversarial modeling framework that consists of a minimax game between two players: the generator and the discriminator. The generator is a multi-layer perceptron G with parameter $\theta^{(G)}$ that intends to create samples to match the real data distribution, $p_{data}(x)$, by mapping the noise vector z, with $z \sim p_z$, to a synthesized sample $G(z;\theta^{(G)})$, which follows a probability distribution $p_g(x)$. The discriminator is another multi-layer perceptron D with parameter $\theta^{(D)}$ that takes in a sample x and outputs $D(x;\theta^{(D)})$ which is the probability that x comes from p_{data} rather than p_g. D is trained to maximize its accuracy of correctly labeling the training samples and the generated samples, while G is trained to weaken D's prediction accuracy. Such interaction results in a two-play minimax game, Eq. (1). This can be computed as the negative cross-entropy of two data batches—one with real data and another with synthetic data from the generator.

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \qquad (1)$$

In subsequent studies, e.g., Radford et al. [15], both *G* and *D* are replaced by a deep convolutional neural network (CNN) to enhance the GAN performance, but the objective function is still consistent with Eq. (1). For convenience, such family of GAN is denoted as ordinary GAN in this paper.

3.2 Semi-supervised *K*+1-class GAN classifier

The ordinary GAN is trained in an unsupervised learning manner, and its discriminator only differentiates unlabeled real samples from those synthesized by the generator. Herein, we increase the output dimension of the discriminator from 2 (“real” or “synthetic”) to *K*+1 such that the discriminator can classify samples from *K* real classes (for samples in p_{data}) and one “generated” class (for samples in p_g). The model can now learn from both unlabeled and labeled data.

The discriminator *D* takes in a sample *x* and outputs a *K*+1-dimensional vector of class probabilities $p_{model}(y|x)$, where $p_{model}(y = i|x) = \exp\,(D(x)_i)/\sum_{j=1}^{K+1} \exp\,(D(x)_j)$ and $\{D(x)_1, \dots, D(x)_{K+1}\}$ are logits corresponding to each class. Herein, $p_{model}(y = K + 1|x)$ represents the predicted probability that sample *x* is “synthetic,” and $D(x)$ in Eq. (1) can be substituted by $1 - p_{model}(y = K + 1|x)$. Similarly, $1 - D(G(z))$ in the second term of Eq. (1) is equivalent to $p_{model}(y = K + 1|x)$. In order to minimize the discriminator loss, value function $V(\cdot)$ in Eq. (1) is reformulated into Eq. (2) with negative signs and the above substitutions, which becomes unsupervised without label information.

$$J^{(D)}{}_{unsupervised} = -\{\mathbb{E}_{x\sim p_{data}(x)} \log[1 - p_{model}(y = K + 1|x)] + \mathbb{E}_{x\sim p_g(x)} \log[p_{model}(y = K + 1|x)]\} \tag{2}$$

The supervised loss of the discriminator is the cross-entropy between *K* real labels and the model’s predicted class distribution $p_{model}(y|x)$:

$$J^{(D)}{}_{supervised} = -\mathbb{E}_{x,y\sim p_{data}(x,y)} \log p_{model}(y = i|x, i < K + 1) \tag{3}$$

Finally, the total discriminator loss is expressed as follows:

$$J^{(D)} = J^{(D)}{}_{unsupervised} + J^{(D)}{}_{supervised} \tag{4}$$

3.3 Semi-supervised generator loss for non-saturating game

From Goodfellow [14], the zero-sum game loss for *G* in Eq. (5) does not perform well in practice, where $V(D, G)$ is defined in Eq. (1).

$$J^{(G)} = -J^{(D)} = V(D, G) \tag{5}$$

The generator’s gradient vanishes when the discriminator fully distinguishes synthetic samples from the real ones, i.e., when $D(G(z))$ becomes 0. Goodfellow [14] proposed a heuristically motivated GAN game loss for the generator as follows:

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_z \log D(G(z)) \tag{6}$$

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., labeling a generated sample $G(z)$ as real. In this study, we drop the constant multiplier, $\frac{1}{2}$, and modify the expression to fit into the semi-supervised scenario with *K*+1 classes as follows:

$$J^{(G)}{}_{heuristic} = -\mathbb{E}_{x\sim p_g(x)} \log[1 - p_{model}(y = K + 1|x)] \tag{7}$$

3.4 Generator feature matching

From Salimans [24], feature matching is a technique that prevents overtraining the generator and increases the stability of GANs. The generator is trained to output samples which yield similar activations at an intermediate layer of the discriminator as those from the real samples as follows:

$$J^{(G)}{}_{feature\ matching} = \left\|\mathbb{E}_{x\sim p_{data}(x)} f(x) - \mathbb{E}_{z\sim p_z(z)} f(G(z))\right\|_2^2 \tag{8}$$

where $f(x)$ is the activations on an intermediate layer of the discriminator for a given sample x. In this study, we choose the flattened output (activated by ReLU [25]) of the last convolution layer of the discriminator network as $f(x)$.

The total generator loss is finally computed as follows:

$$J^{(G)} = J^{(G)}{}_{heuristic} + J^{(G)}{}_{feature\ matching} \tag{9}$$

3.5 The training procedure of the BSS-GAN

In order to address the strong class imbalance issue, a special training procedure using the idea of balanced batch sampling is proposed. Instead of randomly sampling data from the raw dataset to form the training batch, where the class ratio is extremely biased, samples from each class are equally drawn to form the batch with a balanced class ratio. Such strategy may be helpful in eliminating the prior bias during training. This whole pipeline including semi-supervised GAN and balanced batch sampling is named the Balanced Semi-supervised GAN (BSS-GAN). For a training batch size of 2*n*, the detailed procedure is listed below:

i. Initialize the discriminator *D* and the generator *G*.

ii. Half batch of real samples, $\{x_1, \dots, x_n\}$, is passed into the discriminator, and these samples are equally drawn from *K* classes according to the labels. For each $x_i \in \{x_1, \dots, x_n\}$, the discriminator outputs a *K*+1-dimensional probability vector, $v_i = [p_{model}(y = j|x_i),\ \forall j \in \{1, \dots, K+1\}]^T$.

iii. Half batch of random noise vectors, $\{z_1, \dots, z_n\}$, is fed into the generator. For each $z_i \in \{z_1, \dots, z_n\}$, the generator outputs a generated sample, $G(z_i)$.

iv. The generated synthetic samples, $\{G(z_1), \dots, G(z_n)\}$, are passed into the discriminator. For each $G(z_i) \in \{G(z_1), \dots, G(z_n)\}$, the discriminator outputs a *K*+1-dimensional probability vector $u_i = [p_{model}(y = j|G(z_i)),\ \forall j \in \{1, \dots, K+1\}]^T$.

v. Optimize and update the discriminator and the generator's parameters, $\theta^{(D)}$ and $\theta^{(G)}$, respectively, by minimizing Eqs. (4) and (9).

Repeat steps (ii) to (v) until convergence is achieved or designated number of epochs is reached.

4. Experiment setup

In this study, we investigate one basic structural damage detection task in vision-based SHM, namely reinforced concrete cover spalling condition detection, which aims to classify between labels "spalling" (SP) and "non-spalling" (NSP). SP describes the condition where the exterior of a concrete structure peels off, exposing the interior components (mainly reinforcing bars), whereas NSP may be any other conditions, i.e., either the structure is completely intact, or contains minor cracks but without spalling (cover loss). Details about data preparation and experiment settings are presented in the following sub-sections.

4.1 Data preparation

Gao and Mosalam [8] open-sourced a relatively large structural image dataset, namely the PEER Hub ImageNet (ϕ-Net), which includes eight sub-datasets related to scene level, spalling condition, damage state, material type, etc. In this study, the "spalling condition" subset was used as the raw data source, and it was further processed for designed experiments.

Data preprocessing is conducted as: (1) clean the dataset and select images at the pixel level with high visual quality; (2) further select and store the NSP and SP images in a roughly 4:1 ratio to build the experimental dataset, which simulates the class imbalance issue in practice; (3) resize the images to 128×128 pixels; (4) split the data into training and test sets with a 2:1 ratio. A dataset with a total of 2,084 images is built, and label statistics are shown in Table 1. Compared with the ϕ-Net benchmarking experiments [8] and general CV applications, the dataset used in this study is considered imbalanced and under the low-data regime.

Table 1 – Data statistics

Label	**Training**	**Test**	**Split ratio**
NSP	1130	565	2
SP	259	130	2
Class ratio	4.4	4.3	-

4.2 Experiment design and setting

Besides the class imbalance issue, several other topics are also studied in this work. These are (1) the effects of different label ratios in the training set, and (2) the effects of different training pipelines, i.e., conventional training, training with semi-supervised learning, and conventional oversampling by augmenting the minority class with GAN-generated data.

To investigate (1), a conventional multi-layer CNN discriminative classifier, denoted as the baseline (BSL), was compared with BSS-GAN, whose discriminator shares the same architecture with BSL (except the last Softmax layer). The details of the network configuration are presented in Section 4.3. Four magnitudes, namely 10%, 25%, 50%, and 100% of the training data, were used for supervised training in both the BSL and BSS-GAN. It should be noted that the labeled images were selected randomly, but the class label ratio (4:1) was still maintained. Moreover, for BSS-GAN training, the remaining data was treated as unlabeled and also fed into the network for semi-supervised training along with the labeled data. To control the experimental settings, both models were trained for 500 epochs with a batch size of 64. The optimizer is Adam [26] with an initial learning rate (LR) of $2{\times}10^{-5}$. The evaluation metrics herein are the overall accuracy, Eq. (10), and the true positive rate (TPR), Eq. (11), where SP is defined as "positive" and NSP as "negative."

$$Accuracy = \frac{True\ Postive + True\ Negative}{Number\ of\ total\ data} \tag{10}$$

$$TPR = \frac{True\ Postive}{True\ Postive + False\ Negative} \tag{11}$$

As for (2), an ordinary DCGAN model was trained for generating synthetic SP images, which shares the same network architecture with BSS-GAN (except the last Softmax layer and the loss function as discussed in Section 3). Only the SP training images were fed into the model. Total training epochs was 80,000 by Adam optimizer with the same LR= $2{\times}10^{-5}$. A total of 871 synthetic SP images were manually selected from random sampling to augment the raw SP images. Finally, the dataset was oversampled as a balanced dataset, where the number of training data for both classes is 1,130. In this experiment, comparisons were made between BSL, BSL with GAN-based oversampled balanced dataset (denoted as GAN-Aug), and BSS-GAN with 100% labeled training data. The experiments were conducted on the TensorFlow platform on and performed on CyberpowerPC with single GPU (CPU: Intel Core i7-8700K@3.7GHz 6 Core, RAM:32GB and GPU: Nvidia Geforce RTX 2080Ti).

4.3 Network architecture

For a fair comparison, the discriminator of BSS-GAN uses the same architecture as BSL, e.g., number of layer and filter, filter sizes, etc., except the last Softmax layer, and its configuration is summarized in Tables 2 and 3. According to findings in Radford et al. [15] and Gao et al. [12], Leaky ReLU [27] was used as the activation function with $\alpha = 0.2$, and batch normalization (BatchNorm) [28] was inserted after intermediate convolutional layers (Conv) with momentum 0.8. To avoid overfitting, dropout [29] was also applied with a 0.25 dropout rate. Similarly, the same generator architecture was used for both BSS-GAN and GAN-Aug, except the loss functions, where for the former we used Eq. (9) and for the latter we used the ordinary generator loss in [13,14]. Because the real image size is only 128×128, a conventional 100-dimensional noise vector was generated from normal distribution as the input to the generator. There is no dropout in the generator, but BatchNorm layers with momentum 0.8 were added after deconvolution (Deconv) layers except for the last one.

Table 2 – Configuration of the discriminator

Layer	**Filter size (#)**	**Activation**	**Shape**	**Notes**
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, 32×32×64)	
Fc-layer	-	Softmax	(N, c)	BSS-GAN: c = 2+1; BSL: c = 2

Table 3 –Configuration of the generator

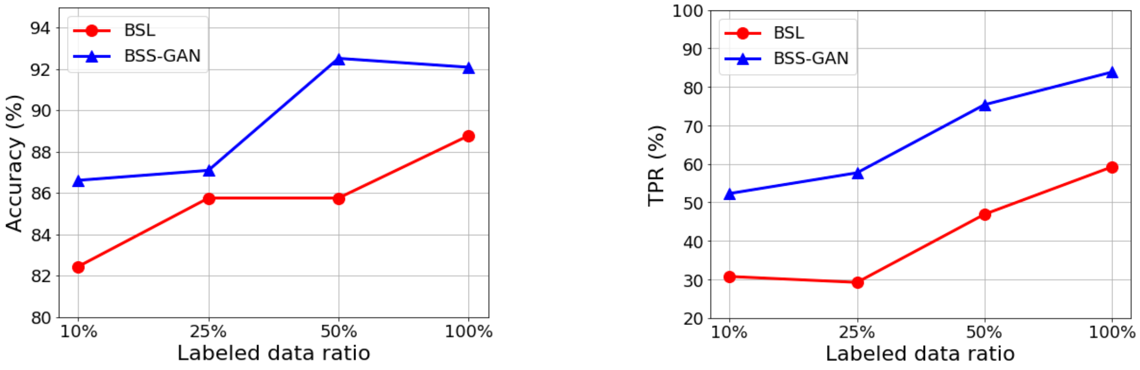
Layer	**Filter size (#)**	**Activation**	**Shape**	**Notes**
Input	-	-	(N, 100)	Random noise from Normal distribution
Fc-layer	-	ReLU	(N, 32×32×128)	
Reshape	-	-	(N, 32, 32, 128)	
Deconv	3×3 (128)	ReLU	(N, 64, 64, 64)	Stride = 2
BatchNorm	-	-	(N, 64, 64, 64)	Momentum = 0.8
Deconv	3×3 (64)	ReLU	(N, 128, 128, 3)	Stride = 2
BatchNorm	-	-	(N, 128, 128, 3)	Momentum = 0.8
Deconv	3×3 (3)	Tanh	(N, 128, 128, 3)	Stride = 1

5. Results and Analysis

5.1 Study on labeled data magnitude

As the main purpose of the study, we investigate the classification performance of BSS-GAN compared with BSL under varying amounts of labeled training data, and the results are shown in Table 4 and Fig. 1. Fig. 1 shows that as the number of labeled data increases, both models have an increasing trend, but BSS-GAN outperforms BSL for all cases. Due to the strong class imbalance in the experiment dataset, simply guessing all data as NSP leads to 81.29% overall accuracy. When only using low magnitudes (10% and 25%) of labeled data, even though both models' accuracies are slightly over 81.29%, the results are not satisfactory. From Table 4, it is observed that the TPR of BSS-GAN is just over 50% and BSL is much worse, which only detected 30% of SP images. It indicates that both models have strong biases to the majority NSP class. At 50% labeled ratio, BSS-GAN has improved performance with 6.76% over BSL in overall accuracy and 75.38% in TPR. At 100% labeled ratio, the overall accuracy of BSS-GAN maintains the same level at around 92% (ignoring the 0.43% drop due to numerical randomness), but the TPR of BSS-GAN reaches 83.85%, which is significantly higher than BSL. In other words, while feeding more labeled data to BSS-GAN, it is more sensitive and robust to detect the minority SP class with the trade-off of a slight accuracy decrease in NSP (the model is more conservative by treating more NSP as SP). Such performance enhancement is meaningful and beneficial to practice, since high TPR and overall accuracy indicate a safer and more conservative model in SHM.

Even though both BSL and BSS-GAN share the similar discriminative part, the dynamic data augmentation hidden in BSS-GAN (which originates from the interaction between the discriminator and the new synthetic data produced by the generator during training) and the contribution from the unlabeled images for feature learning significantly enhance the performance of the BSS-GAN's discriminator. In conclusion, the results validate the effectiveness of using GAN with a semi-supervised learning mechanism under the conditions of data deficiency and strong class imbalance, which alleviates the bias issue faced by conventional CNN classifiers.



(a) (b)

Fig. 1 – Trends of BSS-GAN vs BSL for increasing amounts of labeled data. (a) Accuracy. (b) TPR.

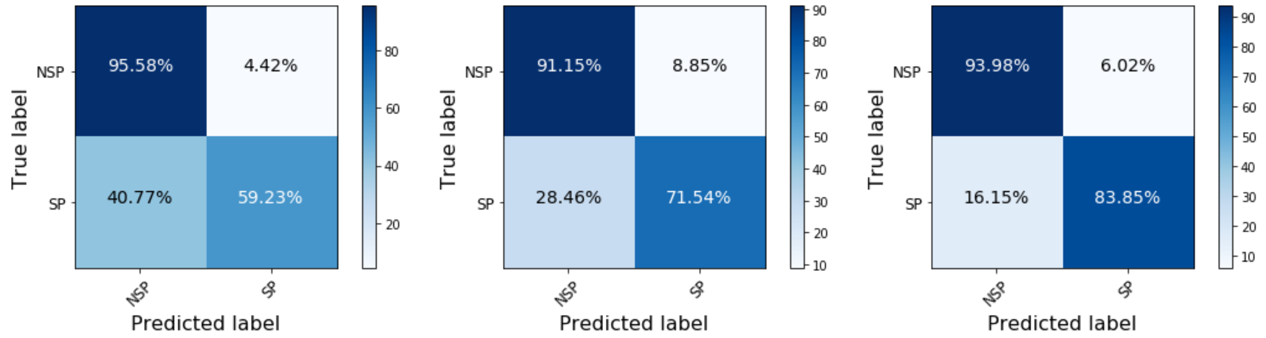
Table 4 – Classification performance of BSS-GAN and BSL for varying amounts of labeled data (%)

Metrics	Model	Ratio of labeled data			
		10%	**25%**	**50%**	**100%**
Accuracy	BSL	82.44	85.76	85.76	88.78
	BSS-GAN	86.62	87.10	**92.52**	92.09
TPR	BSL	30.77	29.34	46.92	59.23
	BSS-GAN	52.31	57.69	75.38	**83.85**

5.2 Study on training pipeline

Based on the experiments, using the GAN-augmented dataset, the overall accuracy of GAN-Aug is 87.49%, and the TPR is 71.54%. Furthermore, plots of the confusion matrix (CM) with respect to three pipelines are presented in Fig. 2. Comparing with BSL and BSS-GAN, the overall accuracy of GAN-Aug is lower than both and the TPR is between the two. This observation can be partially explained through using a balanced dataset where the model is less likely to be trained biased towards a single class, as opposed to BSL which is trained on a strongly biased training set. Similar to BSS-GAN, the increased accuracy in detecting SP (TPR) makes GAN-Aug less sensitive to the NSP images. However, the accuracy improvement in SP does not compensate the accuracy decrease in NSP, which causes a decrease in the overall accuracy. In other words, even though the GAN-Aug classifier achieves a better TPR than BSL, it loses more capability to recognize the NSP images, which does not make it a better model than BSL. Therefore, this experiment provides another example showing that simply oversampling the raw dataset with GAN-generated data may not work as stated in Gao et al. [12]. On the contrary, in this experiment, BSS-GAN achieves both higher overall accuracy and TPR than the other two, making it a state-of-the-art GAN-related pipeline.

As for computational efficiency, the training time costs for BSL, BSS-GAN and GAN-Aug with the full dataset are approximately 0.75 hour, 1.5 hours and 2 hours, respectively. In general, BSL spends the least time, but considering its poor performance, this value is not contrastively meaningful. Compared to BSL, BSS-GAN needs to train extra parameters in the generator and the generator also needs additional computational resources to synthesize images from the noise vectors. These additional operations demand extra time, which is still acceptable. Especially when compared with GAN-Aug, BSS-GAN speeds up by 25%, which indicates its high potentials in efficiency improvement over other GAN-based methods.



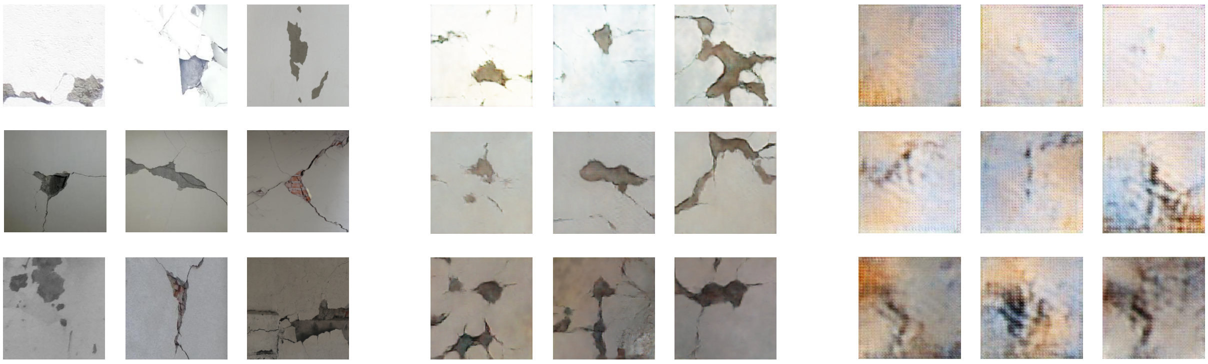
(a) (b) (c)

Fig. 2 – Confusion matrix of different pipelines. (a) BSL. (b) GAN-Aug. (c) BSS-GAN.

5.3 Study on synthetic image quality

In this section, synthetic images generated by both BSS-GAN and GAN-Aug are shown in Fig. 3. In general, no model collapse issues occurred in either result and the synthetic images have good visual variety. However, the visual quality of images generated by GAN-Aug, Fig. 3(b) is much better than that of the BSS-GAN, Fig. 3(c), and this observation can be explained by several factors.

Unlike the mechanisms in ordinary GAN, the loss function in BSS-GAN focuses more on classification than image generation (reflected by the supervised cross-entropy loss); on the contrary, the ordinary GAN only includes the unsupervised loss, which is more about feature learning than classification. The different training objectives influence the performance of the generator even though the same generator architecture is used. Therefore, it can be inferred that the improvement in the discriminator's classification abilities may weaken the generator when the classification task of the discriminator increases from binary to K+1 classes.



(a) (b) (c)

Fig. 3 – Sample images from (a) real SP images, (b) GAN-Aug generated images, and (c) BSS-GAN generated images.

According to Gao et al. [12], structural images have a complex and mixed distribution, which makes ordinary GAN hard to generate clear and class-discriminative images. However, through conditioning, e.g., considering class information related to a specific class of images makes ordinary GAN generate higher quality images towards that class. In GAN-Aug pipeline, only feeding pixel level SP images during GAN training can be viewed as one conditioning operation, which significantly reduces the complexity of the data distribution. Thus, in Fig. 3(b), the synthetic SP images show very realistic visual qualities. On the contrary, the generator in BSS-GAN was trained with both NSP and SP images in an unsupervised manner, thus it has to learn a mixed distribution from both NSP and SP. As a result, synthetic images generated by BSS-GAN can be viewed as either in NSP, SP or the intermediate state. To show this, in Fig. 3(c), synthetic images in the first row are smooth and resemble NSP, and the remaining images show some blurry, damage-like features, e.g., cracks and dark spots.

In summary, if only focusing on the synthetic image quality, ordinary DCGAN with class conditioning has a better image generation ability than BSS-GAN under 128×128 pixels. But according to the experimental feedbacks in classification performance, it can be inferred that even though synthetic images by BSS-GAN are not clear enough and less realistic, the learned features and patterns from these generated images may still contribute to the learning process of the discriminator in the semi-supervised learning pipeline, improving both the training efficiency and the discriminative performance.

6. Conclusions and Extensions

In this study, we firstly pointed out two key issues impeding the application of DL in vision-based structural damage assessment, namely data deficiency and class imbalance. As an alternative method to TL and conventional DA, GAN-based augmentation was discussed. In order to improve the efficiency of using GAN in the classification task, a semi-supervised learning GAN pipeline along with the balanced batch sampling technique was introduced, named BSS-GAN. Under the imbalance-class and low-data regime, a series of computer experiments with respect to reinforced concrete cover spalling detection, which is one of the most basic damage assessment tasks in vision-based SHM, were designed and conducted.

To investigate the discriminative capability, under different magnitudes of labeled data, BSS-GAN was compared with BSL, which shares the same discriminative network architecture. It is observed that BSS-GAN outperforms BSL in both overall accuracy and TPR at all labeled ratios (especially at 50%), which validates the effectiveness of BSS-GAN under the above-mentioned constraints. Furthermore, using the whole training dataset as labeled data, BSS-GAN was compared with another GAN-based pipeline, namely GAN-Aug, which oversamples the minority class (SP) with synthetic data generated by an ordinary DCGAN until reaching the same amount of majority class (NSP), and trains the BSL with this balanced dataset. The results indicate that GAN-Aug improves the TPR compared with BSL by using more SP images, but it is still ~10% lower than BSS-GAN. Moreover, it sacrifices the performance in detecting NSP image, which leads to the decrease of the overall accuracy (which is far lower than BSS-GAN). Therefore, in our experiments, BSS-GAN achieves the state-of-the-art classification performance over both BSL and GAN-Aug.

In our preliminary study, there still exist several factors that need to be explored. By computer experiments, BSS-GAN was only shown to be effective under the low-data regime, more experiments need to be conducted for medium-data regime or even large datasets, which can be realized by continuing the efforts in collecting and labeling structural images in practice. Moreover, only one CNN architecture was tested in this paper, and more parametric studies with respect to network architecture are expected.

In conclusion, the presented promising results by BSS-GAN shed light on the high potentials of semi-supervised GAN in vision-based damage assessment and SHM. This is clearly worth significant explorations and research efforts in the future.

7. Acknowledgements

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

8. References

[1] 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), 361-78.

[2] Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J. & 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.

[3] Gao, Y., Li, K., Mosalam, K., & Günay, S. (2018). Deep residual network with transfer learning for image-based structural damage recognition. *11th US National Conference on Earthquake Engineering, Integrating Science, Engineering & Policy*, Los Angeles, USA.

[4] 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.

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

[6] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Technical report*, University of Toronto.

[7] Pan, S. J. & Yang, Q. (2010), A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345-59.

[8] Gao, Y. & Mosalam, K. M. (2019). PEER Hub ImageNet (*Φ*-Net): A large-scale multi-attribute benchmark dataset of structural images, *Technical Report* PEER 2019/07, Pacific Earthquake Engineering Research, Berkeley, USA.

[9] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Li, F. (2009). Imagenet: A large-scale hierarchical image database. *In Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*. 248-255.

[10] Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.

[11] He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition. *In Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, Las Vegas, NV, 770-78.

[12] 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.

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative adversarial nets. *In Advances in neural information processing systems*. 2672-2680.

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

[15] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial net- works. arXiv preprint arXiv:1511.06434.

[16] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:1701.07875.

[17] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

[18] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

[19] 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*. 1125-1134.

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

[21] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using GAN for improved liver lesion classification. *In 2018 IEEE 15th International Symposium on Biomedical Imaging*. 289-293.

[22] Madani, A., Moradi, M., Karargyris, A., & Syeda-Mahmood, T. (2018). Chest x-ray generation and data augmentation for cardiovascular abnormality classification. *In Medical Imaging 2018: Image Processing*. International Society for Optics and Photonics.

[23] Madani, A., Moradi, M., Karargyris, A., & Syeda-Mahmood, T. (2018). 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*. 1038-1042

[24] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. Advances in Neural Information Processing Systems, 2234-2242.

[25] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *In Proceedings of the 27th International Conference on Machine Learning*, 807–814.

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

[27] Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.

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

[29] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, **15**, 1929-58.