Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-020-01710-x

Synthetic data augmentation for surface defect detection and classification using deep learning

Saksham Jain[1] · Gautam Seth[1] · Arpit Paruthi[1] · Umang Soni[1] · Girish Kumar[2]

Received: 29 October 2019 / Accepted: 29 October 2020

Abstract
Deep learning techniques, especially Convolutional Neural Networks (CNN), dominate the benchmarks for most computer vision tasks. These state-of-the-art results are typically obtained through supervised learning, for which large annotated datasets are required. However, acquiring such datasets for manufacturing applications remains a challenging proposition due to the time and costs involved in their collection. To overcome this disadvantage, a novel framework is proposed for data augmentation by creating synthetic images using Generative Adversarial Networks (GANs). The generator synthesizes new surface defect images from random noise which is trained over time to get realistic fakes. These synthetic images can be used further for training of classification algorithms. Three GAN architectures are trained, and the entire data augmentation pipeline is implemented for the Northeastern University (China) Classification (NEU-CLS) dataset for hot-rolled steel strips from NEU Surface Defect Database. The classification accuracy of a simple CNN architecture is measured on synthetic augmented data and further it is compared with similar state-of-the-arts. It is observed that the proposed GANs-based augmentation scheme significantly improves the performance of CNN for classification of surface defects. The classically augmented CNN yields sensitivity and specificity of 90.28% and 98.06% respectively. In contrast, the synthetically augmented CNN yields better results, with sensitivity and specificity of 95.33% and 99.16% respectively. Also, the use of GANs is demonstrated to disentangle the representation space and to add additional domain knowledge through synthetic augmentation that can be difficult to replicate through classic augmentation. The proposed framework demonstrates high generalization capability. It may be applied to other supervised surface inspection tasks, and thus facilitate the development of advanced vision-based inspection instruments for manufacturing applications.

Keywords Surface defects · Classification · Convolutional neural network · Generative adversarial network · Deep learning

Introduction

In today's world, the industry is facing a challenge in the use of supervised learning algorithms for classification. Despite the various advantages of these learning algorithms, their performance decreases when trained on a real-world dataset. One reason is a shortage of training data due to factors that are typical to the manufacturing industry. With the advancements made in modern manufacturing processes and implementation of quality programs including six sigma, the number of good samples greatly exceeds the number of defective samples. The number of defectives are sometimes to the tune of parts per million (ppm). Therefore, a class imbalance exists in manufactured samples. However, if we discard the surplus of non-defective samples in order to keep class balance, the total dataset size is drastically reduced. Thus, to create a large dataset of defects, there is an opportunity cost involved due to having to manufacture in large numbers without addressing the defects and their causes. Therefore, manufacturing defect datasets are typically smaller.

Industrial anomaly detection is a highly challenging task because what constitutes a defect and what does not, can be considered a soft boundary, which can lead to false-negatives or false-positives depending upon the priority of classification. In addition to the shortage of the sample data, the available datasets can contain multiple categories of defects inside their positive (defective) class which can be difficult to model. One method to handle this classification problem is to train the classifier only on the positive (defective) sam-

✉ Umang Soni
umangsoni.1@gmail.com

[1] Netaji Subhas University of Technology, New Delhi, India

[2] Delhi Technological University, New Delhi, India

Published online: 18 November 2020

ples. In such a case, even if the data contains multiple classes of defects, there are no non-defective samples. This leads to a cautious classification boundary, as only when a sample does not correspond to any class of defects with a high enough confidence, is it classified as negative (non-defective) (Lai et al. 2018). However, this approach is still unable to handle the problem of small dataset size, which is particularly important when working with Convolutional Neural Networks (CNNs). The standard solution is the augmentation of the training data by rotating, reflecting, cropping, translating and scaling existing images to increase the dataset size and include some additional domain knowledge such as rotation-invariance, size-invariance, etc.

Deep neural networks have lower performances on small datasets when trained using convolutional techniques in comparison to traditional machine learning methods such as the support vector machine (SVM). But it has been experimentally proven that deep networks outperform such methods when trained on larger datasets (Feng et al. 2019). A significant reason for this parity is that SVMs rely on handcrafted features while deep neural networks are purely data-driven and as such, are better able to extract and utilise salient features from larger datasets for the classification task (Tabernik et al. 2020).

In this paper, authors investigate the use of GANs to sample additional synthetic data by modeling the training data distribution for manufacturing datasets. This synthetic data can be used for augmentation of the real data that is subsequently used to train the CNN for classification. Although GANs cannot produce images with perfect fidelity despite recent advancements, the authors of (Dosovitskiy et al. 2015; Richter et al. 2016) demonstrate marked improvements in results after synthetic augmentation of the data even with poorer fidelity images. Thus, expanding the dataset with these realistic, if slightly degraded, synthetic samples helps in reducing overfitting and improving the accuracy as well as the generalization capability of the CNN classifier (Madani et al. 2018). Authors verify these assertions and evaluate the suitability of the entire pipeline for real-world applications.

In Frid-Adar et al. (2018), the authors demonstrate promising results for medical image classification tasks by employing the use of GANs for data augmentation directly as a pre-processing step without any additional data. In this paper an attempt has been made to further build upon these works for surface defect detection and classification tasks.

Motivation and contribution

Apart from the typically smaller size of manufacturing datasets used in defect detection and classification tasks, two other specific problems occur. Features present in the intra-class defects may be random and highly diverse whereas the features in the inter-class defect may be nearly identical (Song and Yan 2019). Also, the defect images can vary because of the ambient illumination, slight material changes, and the severity of the defect that lead to a change in the grey value. This change can have an effect on the stability of the obtained defect features and may result in a reduction in the recognition rate.

While the application of GANs to expand the dataset with realistic but distinct fakes of the training data is straightforward, it is capable of handling the above two issues. Hand-crafted features are not required for deep learning algorithms. GANs also do not depend upon the user and instead model all the sources of variation (features) in the dataset, by learning the distribution of the training samples. Thus, as long as distinct features exist, even if they are too distributed for intra-class defect samples or too similar for inter-class defect samples, GANs have the potential to capture and model them correctly.

Data pre-processing involves transformation of the discrete distribution of training samples into a continuous distribution. It allows for GANs to model even challenging features present in the dataset and generate examples along the continuum. For example, if samples with different levels of severity of defects are used, the GAN would learn to distinguish between them and generate samples of all levels of severity, though they are not explicitly present in the training data. With recent advances made in GAN architectures (Chen et al. 2016), it has become possible to control the latent space of the generator which captures such information. Thus, if sufficient samples of the features affecting the grey values are generated, a more robust classifier can be trained.

Following the work of Madani et al. (2018) and Frid-Adar et al. (2018), the use of GANs as a pre-processing step for manufacturing tasks is proposed in this study. Further, the use of GANs for disentangling the representation space of the training data is explored. Thus, the objective of the paper is three-fold:

- To evaluate the efficacy of GAN architectures for synthetic augmentation of small manufacturing datasets, and to investigate GAN-based procedure as a standard augmentation technique during pre-processing.
- A simple CNN trained on synthetically augmented data is shown to outperform more complex CNN based feature extractors trained with classic augmentation.
- To explore and disentangle the latent space of the training data for obtaining a degree of control on the additional domain knowledge locked within the dataset that cannot be obtained through traditional classical augmentation techniques.

An illustration is provided to demonstrate the proposed augmentation and CNN classification pipelines and to evaluate their performance on the NEU-CLS dataset of surface

defects in hot-rolled steel strips. The NEU-CLS dataset contains 6 balanced classes of defective samples but no non-defective samples. Further details of the dataset are given in "Surface defects in hot-rolled steel strip" section.

Literature survey

Generative adversarial networks were first introduced in Goodfellow et al. (2014). They implemented discriminator and generator which were based on multi-layer perceptrons and were trained using backpropagation.

GANs have gained popularity in recent times, as numerous variations of GANs have been developed, which further improve the quality of the generated images and also widen the scope for countless applications. Deep convolutional generative adversarial network (DCGAN) constitutes a constrained architecture that reduces the gap between CNNs for supervised and unsupervised learning (Radford et al. 2015). Least square generative adversarial network (LSGAN) uses least square as a loss function for the discriminator which leads to higher quality image generation and more stable training (Mao et al. 2017). In a conditional generative adversarial network (CGAN), the discriminator and generator were conditioned to generate images based on class labels (Mirza and Osindero 2014). Conditional deep convolutional generative adversarial network (C-DCGAN) is a combination of DCGAN and CGAN. C-DCGAN utilizes convolutional neural network for feature extraction and with a conditional extension to augment data (Luo et al. 2020). The InfoGAN is based on an information theoretic formulation that allows for manipulation of the latent space (Chen et al. 2016). Also, InfoGAN uses a sleep–wake algorithm for training (Chen et al. 2016). Wasserstein generative adversarial network (WGAN) improves the training algorithm to avoid problems of mode collapse and provides better learning curves (Arjovsky et al. 2017). Auxiliary classifier generative adversarial network (ACGAN) employed label conditioning (Odena et al. 2017) while energy based generative adversarial network (EBGAN) involves the use of energy functions for discriminator and generator as a producer of low energy samples (Zhao et al. 2017). Boundary seeking generative adversarial network (BGAN) trains the discriminator with a heuristic of difference between target and resulting images (Hjelm et al. 2018) and boundary equilibrium generative adversarial network (BEGAN) uses an equilibrium based method on top of WGAN (Berthelot et al. 2017).

Some of the key applications of GANs include handwriting recognition (Tian 2017), facial age progression (Zhang et al. 2017), super-resolution imaging (Ledig et al. 2017), visual saliency prediction (Pan et al. 2017), object detection (Li et al. 2017), and unsupervised domain adaptation (Bousmalis et al. 2017). Owing to their excellent results in image synthesis, researchers have begun using GANs for data augmentation. These have been used initially to improve image quality and then for further training (Shrivastava et al. 2017), for generating different versions of an image (Antoniou et al. 2018), and even for imposing emotions on bland faces to augment smaller classes (Zhu et al. 2018). GANs have also find application in data augmentation for medical imaging tasks such as denoising CT scans (Wolterink et al. 2017), medical image segmentation, brain MRI segmentation and synthetic augmentation (Moeskops et al. 2017), and image segmentation of the liver (Yang et al. 2017), of skin lesions (Izadi et al. 2018) and of the brain (Alex et al. 2017).

In the manufacturing industry, computer vision is widely used with applications in, magnetic tile surface defects (Huang et al. 2018), additive manufacturing anomaly detection (Scime and Beuth 2018; Davtalab et al. 2020), steel surface defect inspection (Hao et al. 2020; Sun et al. 2016), defect detection in lithium ion battery electrodes (Badmos et al. 2020), roughness prediction (Grzenda and Bustillo 2019) etc. GANs models, especially after improvement of its training algorithm (Gulrajani et al. 2017), have been proposed for anomaly detection in structured and arbitrary textured surfaces (Lai et al. 2018) and unsupervised inspection of surfaces (Zhai et al. 2018).

Surface defects in hot-rolled steel strip

Steel-strip, being one of the most elemental components in the steel industry, has a high impact on the characteristics of the end product. Due to limitations in manufacturing processes, it encounters various defects like inclusion, surface defects, etc. (Song and Yan 2013). These surface defects can be caused due to many factors, which can be broadly divided into the following two categories:

- Defects on the continuous slabs introduced during the rolling process.
- Defects due to inappropriate rolling technique or foreign body inclusions in the strip-roll interface (Devadas et al. 1991; Yu et al. 2013).

Surface defect detection and classification

The main purpose of anomaly detection is to find out deviations or outliers from normal data. It has many industrial applications, from system health monitoring (spotting a malignant tumour in an MRI scan), to fault detection in manufacturing processes (through visual inspection of castings or welds). In the case of steel strips, the deviations can be in the form of surface defects. Many computer vision approaches based on feature extraction utilizing deep learning have been used for anomaly detection and classification (Lai et al. 2018). Owing to the data requirements

of deep learning models, and the typically small size of manufacturing datasets (and sometimes class imbalance), data augmentation is required. Basic data augmentation techniques can be graphical transformations such as translation, rotation, scaling, mirroring, etc. These may not be able to add sufficient information to the dataset after a certain limit. Hence, generative models like generative adversarial networks (Goodfellow et al. 2014), variational autoencoder (Kingma and Welling 2014), restricted Boltzmann machine (Carreira-Perpiñán and Hinton 2005), Helmholtz machine (Hinton et al. 1995) can be employed. Generative Adversarial Networks (GANs) is selected for this work. GAN consists of two convolutional neural networks, namely Generator and Discriminator. These are trained in an adversarial process, in which the generator generates fake images while the discriminator distinguishes between the fake and real images. GANs have recently gained popularity since good quality synthetic images can be generated using them. Detailed discussion on GANs is provided in "Synthetic augmentation" section.

Further, the details of the augmentation process for the surface defect dataset are provided in "Data augmentation" section, while for the detection and classification process we use the "Convolutional neural network" architecture as described in subsection.

Dataset

The dataset used to evaluate the effectiveness of the proposed model was generated by Northeastern University (NEU). This dataset includes six frequent hot rolling defects for steel strips: Rolled-in scale (RS), Patches (Pa), Crazing (Cr), Pitted Surface (PS), Inclusion (In) and Scratches (Sc).

Data set consists of a total of 1800 sample images for the above-mentioned defects, i.e., 300 sample images for each defect, with a resolution of 200×200 pixels and a grayscale value from 0 to 255. In this study, the intra-class defects have been defined as those having the same kind of defect while the inter-class defects are those having the different classes of the defect. Intraclass defects can be easily observed in Fig. 1, for example, different orientations of cracks. The inter-class defects may have identical aspects, for example, in RS, Cr, PS. Also, the defect images can vary due to illumination and material changes, (Song and Yan 2019).

Classification

Classification refers to a machine learning technique used to determine the class or categorization that a given object belongs to. The output of a classification algorithm is a discrete response with the predicted class as the output. Any architecture or algorithm for classification tasks is said to work by observing the labeled samples in the given training set, and then uses a set of criteria (whether rule-based or by learning over time) to attempt drawing conclusions in order to classify the samples in the test set. Provided with one or more inputs, a classification model attempts to predict the value (e.g. class labels) of one or more data samples (e.g. surface defects).

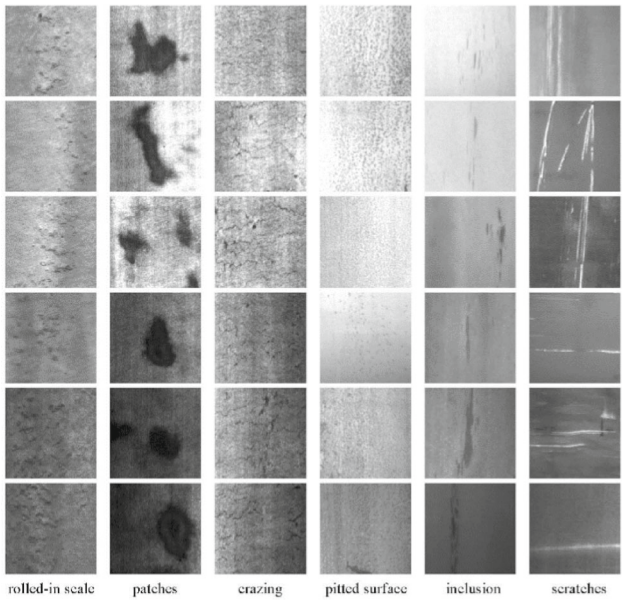


Fig. 1 Inter and Intraclass surface defects (Song and Yan 2019)

In this work, effort has been made to classify surface defects in hot-rolled steel strips using a Convolutional Neural Network-based deep learning approach. This approach is employed as the CNN has powerful architectures for image classification tasks, and the way forward for future applications (Deng et al. 2009). However, CNN typically requires massive amounts of data. Therefore, CNN-based classification becomes feasible for this study as the NEU-CLS dataset can be appreciably enlarged via data augmentation. Refer "Data augmentation" section for detailed discussion.

Convolutional neural network

In fully connected neural networks (FNNs), each neuron or node in all layers (disregarding the input layer), is connected to every neuron in the preceding layer. Matrix multiplication is utilized to obtain the weights and biases for each neuron, and there are thousands and thousands of such neurons in a deep network. This makes the use of FNNs not only very computationally intensive but also requires a large amount of time to successfully train them. To overcome this, CNN was proposed as an extension to standard deep neural networks. CNN uses an operation known as a convolution in their layers rather than matrix multiplication, and as a result, substantially decreases the number of parameters to train in the network, along with achieving a faster forward-propagation process

that makes their training and implementation faster. This is advantageous because the learned convolutional filters are consequently used for the entire image. Another major reason for their widespread use and success in most computer vision tasks is that all optimization techniques (e.g. Backpropagation with stochastic gradient descent) used in standard FNNs are also applicable for CNNs.

Implementation details for baseline

The CNN which was used for classification had 3 convolutional layers. The first and second convolutional layers had 32 filters of size 3×3 while the third convolutional layer had 64 filters of size 3×3. The Rectified linear unit (ReLU) activation function was used for all the layers. The fully connected layer had 256 neurons. The model was trained for images with 3 colour channels, i.e., Red Green Blue (RGB). In the CNN, Adam optimizer is used with softmax cross-entropy with logits as the cost function. The optimizer had a learning rate of 0.0001. For training and validation, a train-test split of 9000 images and 1800 images respectively was used while keeping the classes balanced. During the testing time, the same validation set was utilized for all experiments. The data consists of 6 classes with the mapping: Cr: 0, In 1, PS: 2, Pa: 3, RS: 4, Sc: 5 which was used for classification. The training of the model was done in batches of size 60 for 20 epochs (Fig. 2).

Implementation details for fine tuning

The above architecture was pre trained on a smaller subset[1] of the imagenet dataset (Deng et al. 2009) and consequently fine-tuned on the actual 9000 training images and 1800 validation images, keeping the training conditions the same.

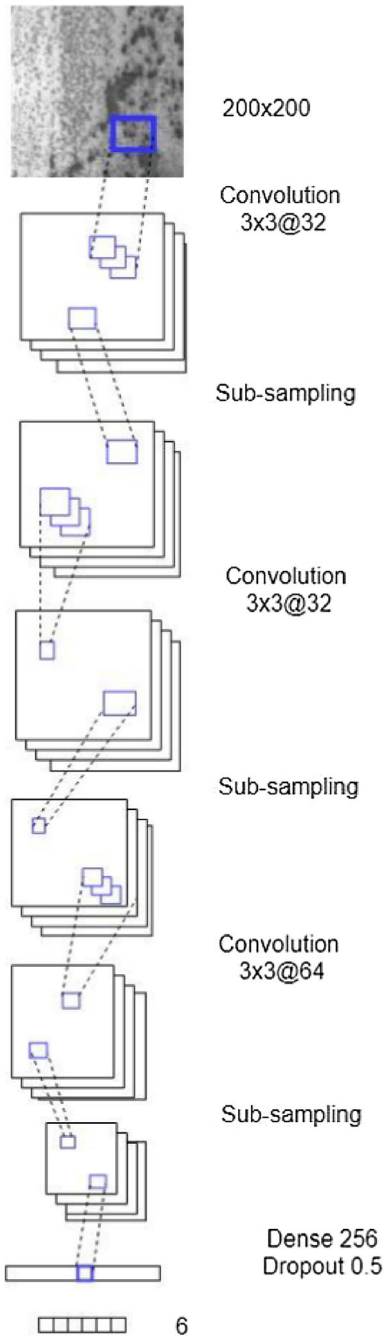


Fig. 2 Architecture of the proposed CNN

Data augmentation

Data augmentation is a procedure carried out to enlarge a small input training set to get more effective results. Most important information can be extracted from the small training set when it is subjected to the augmentation process, because it allows us to include some knowledge about potentially never-before-seen data during training, without having to collect additional data. For example, if the end application is such that image features can appear in any orientation during the testing phase, it would be useful to introduce randomly rotated samples of the existing dataset into the training procedure.

In the data augmentation operation, the additional input training samples can be created by two methods: (i) geometrically transforming (usually, affine transformations are performed) the given training samples to increase input variety, in a process known as classic transformation, or (ii) generating new never-before-seen samples using generative models trained on the existing input training sample, in a process known as synthetic data augmentation.

To be successfully benefitted from data augmentation techniques, it is important to keep in mind the details and specifics of the dataset, the network framework, and the task, because haphazard or unfit augmentation rules and opera-

tions may lead to even worse performance of the model than if the dataset had not been augmented.

The following sections, first the classic augmentation operation along with its implementation has been described and then our novel approach to synthetic augmentation using generative adversarial networks for surface defect detection and recognition task has been illustrated.

Classic augmentation

Thousands of parameters must be trained for even small CNNs. When deep learning architectures with multiple layers are built, or when the training set is limited in terms of the number of images, there is a high likelihood of overfitting. One of the standard solutions to resolve the problem of overfitting is to artificially enlarge the dataset via data augmentation. For greyscale images, classic augmentation refers to data augmentation techniques that involve geometric coordinate transformations, e.g. random translation, random rotation, scaling, random flipping, elastic deformation, and shearing. For our particular application—surface defect detection and recognition –elastic deformation or shearing has not been used so as to protect the useful surface defect features or characteristics that depend upon the shape of the image.

Implementation details

The dataset was expanded using the classic data augmentation method. The following transformations were applied to the dataset:

- The images were randomly rotated in clockwise or anticlockwise direction with an angle ranging from 0° to 180°.
- The images were randomly flipped about the vertical or horizontal axes.
- The images were translated by random sampling of [x,y] pixel values in a set range.
- The images were randomly rescaled to a factor between 0.5 to 1.5 times.

Ultimately, all images were resized to the original size of 200×200 pixels by bicubic interpolation. With the help of these transformations, the size of the training set was increased to 9000 images. Three such training sets were generated, and the mean score is reported in "Results and discussion" section.

A major issue that arises with classic augmentation is that the improvement offered saturates after a point (Richter et al. 2016). This is because no new additional information is actually being added with classic augmentation, and after a while, the model becomes robust to the changes brought on by simple transformations since only a limited set of operations can be performed. Thus, to introduce more diversity into the training samples, synthetic data augmentation is required.

Synthetic Augmentation

Synthetic data augmentation differs from classic augmentation where the additional images are synthetically generated fakes, which closely resemble the actual input images rather than being the transformed input images. Although data augmentation for computer vision tasks has long existed, the advent of advanced GANs has provided a unique opportunity for taking it further.

A correctly trained GAN can augment each source of variance in the dataset, simultaneously obtaining a continuous distribution of training samples by transforming the discrete distribution. If a GAN is trained on sufficient samples at different states of rotation, it will learn to generate samples at any state of rotation. Therefore, a GAN trained on classically augmented data can learn to replicate its effects and include additional domain knowledge such as—translation-invariance (e.g. by horizontal or vertical-shift) or size-invariance (e.g. by random resize), etc. However, the true power of GANs comes to light when considering more challenging features such as the severity of, or size of the image area affected by, the defect. It would be extremely difficult to model the shape, impact, and size of the defect and then use just affine transformations to produce augmentations. However, given sufficient samples of different discrete levels of severity, the GAN will infer the model on its own and learn to generate samples along the continuum of severity. Since CNNs are powerful discriminators, any GAN models will require careful and stable training, not to mention correctly tuned parameters to obtain usable results. While there is no good theoretical basis for how to select hyperparameters for training stable GAN models, which remains an open problem, there exists empirically demonstrated heuristics that work in practice. The authors of DCGAN (Radford et al. 2015) provide recommendations for stably training GANs which are considered as the defect starting point. In this paper, authors use the default values for the hyperparameters, as described in the respective GAN implementations to eliminate the need for optimal hyperparameter search. These values are referred to as out-of-box hyperparameters. This reduces the overall complexity of the pipeline and also demonstrates that even without hyperparameter tuning, obtaining synthetically augmented data that outperforms classical augmentation techniques is possible.

In this section, authors describe three approaches to synthetic data augmentation that utilize the improvements to the original GAN framework, namely Deep Convolutional GAN (DCGAN), Auxiliary Classifier GAN (AC-GAN) and Information-theoretic GAN (InfoGAN) for generating image samples for augmentation.

DCGAN

DCGAN was proposed by Radford et al. (2015), aiming to improve the training stability along with the quality of generated images over Goodfellow's vanilla GAN model. DCGAN better incorporates the recently made advances for training CNNs and uses fully-convolutional neural networks in its architecture.

An important additional attribute of DCGAN is that the discriminator can be used as a powerful tool for feature extraction, as demonstrated in the paper (Radford et al. 2015). Its use, especially for unsupervised image classification tasks, achieves performance that is competitive with the state-of-the-art unsupervised models on benchmark problems such as CIFAR10, etc.

Architecture overview Stride convolutions are used rather than pooling layers, and transposed convolutions are used in place of up sampling so as to make both discriminator and generator networks fully-convolutional. Figure 3 illustrates the DCGAN generator framework as proposed by Radford et al. (2015). Random numbers taken from a uniform distribution (which is learned during the training) are sent as input into the generator network, which encodes this noise into a surface defect image, taken as the generator output. The generator framework contains 4 transposed convolutional layers that up sample the image using a filter of the specified size, and a fully connected layer reshaped to the requisite size. The transposed convolutional layer, termed as the deconvolutional layer can extend the image pixels by pushing zeroes into the intermittent locations. Performing the convolution operation across this extended image leads to the generator outputting an image of a larger size. Batch-normalization is added to all convolutional layers of the framework, with the final generator layer being the exception. The DCGAN learning procedure is stabilized by normalizing the responses to contain variance equal to 1, and mean equal to 0 across the complete mini-batch. This also prevents full or partial mode collapse for the generator framework. Figure 4 shows a batch of images for the Inclusion class generated by the trained DCGAN. It is important to note that the discriminator architecture is simply the generator architecture mirrored.

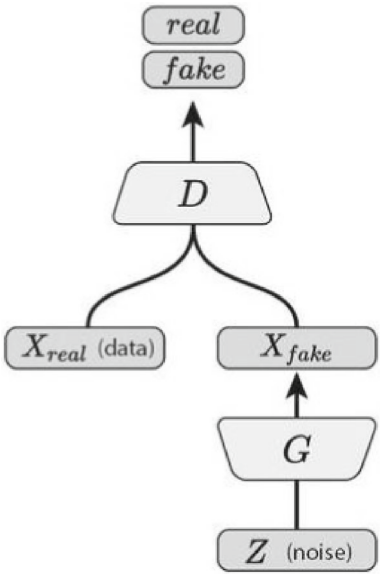


Fig. 3 Diagrammatic representation [figure taken from Frid-Adar et al. (2018)] of the DCGAN Architecture as per the original implementation given in Radford et al. (2015)

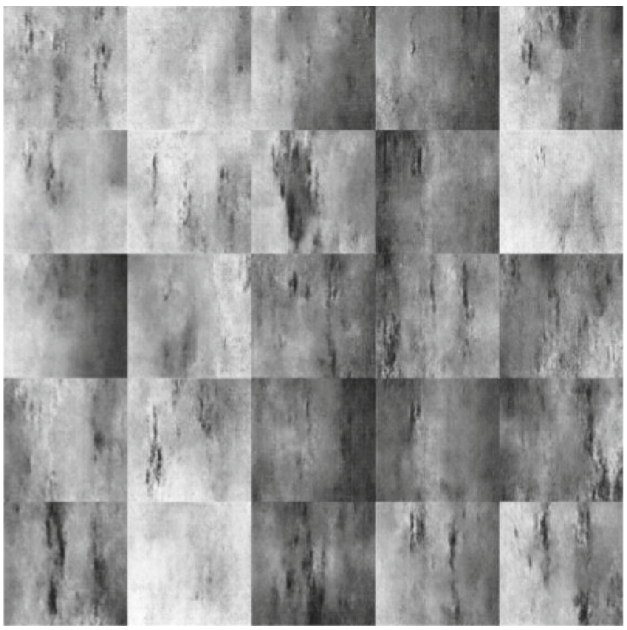


Fig. 4 Images generated by DCGAN

Implementation details The DCGAN was trained separately on the input data in batches of 25 images. The learning rate was set to 0.0002 and momentum (β_1) was set to 0.5. The discriminator of the model had 5 hidden layers, out of which 4 had leaky ReLU as an activation function while the final had a linear activation function. The generator of the network had 3 hidden layers with ReLU as the activation function for all the layers. The DCGAN model was trained for 100 epochs, and 24 synthetic batches of the same batch size were generated for each class. Each batch was generated in the form of a grid of 5×5, which was sliced using the *image_slicer* module available in python. Richter et al. (2016) has been referred to train the DCGAN on a 5400 images-strong randomly selected subset (keeping classes balanced) of the classical augmentation training set, to generate 3600 samples. Three such training sets were generated and the mean score has been reported in "Results and Discussion" section. The size of the resultant training set was kept

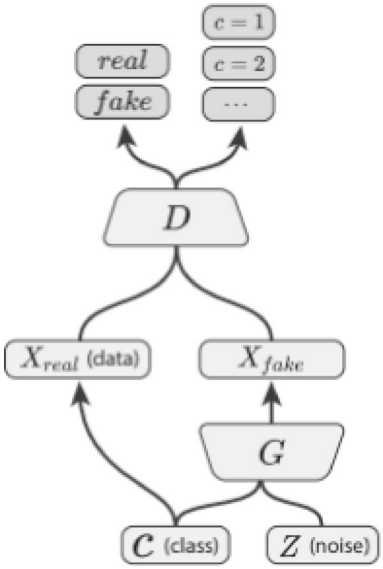


Fig. 5 Diagrammatic representation of the AC-GAN Architecture as per the original implementation given in Odena et al. (2017)

the same so as to fairly compare the results to those of classic data augmentation. DCGAN requires a separate model to be trained in each class, which makes it much more cumbersome to use than classical augmentation. In order to do away with tuning the hyperparameters multiple times, same hyperparameters have been used for all models.

AC-GAN

The second GAN model employed for this study is known as the Auxiliary Classifier GAN (ACGAN) (Odena et al. 2017). It is used for conditional image synthesis and was formulated as an extension to Conditional GANs (Mirza and Osindero 2014). Figure 5 illustrates the typical architecture for this model. ACGAN allows model conditioning on information obtained from an external data source in order to increase the generated image sample quality. Additionally, it is perfectly capable of incorporating the provided class labels from the training set and thus produces labeled image samples. This is useful for synthetic augmentation of datasets for supervised learning tasks, e.g. surface defect detection and recognition tasks.

Architecture Overview The architecture of ACGAN does not appear much different from Goodfellow's GAN. However, ACGAN has major differences in the training procedure and function performed by the discriminator and generator. The ACGAN discriminator is best served by being assigned the task of reconstruction of side information rather than simply inputting the side information into the framework (Richter et al. 2016). To carry out this, the discriminator is so adjusted as to consist of an auxiliary decoder neural architecture. Consequently, it not only provides a decision on whether the image is real or fake but also proffers the class label as an output. The ACGAN is employed in this work for synthetic data augmentation requirements with only slight modifications for the six classes. The auxiliary decoder in the discriminatory framework successfully classifies all six classes of surface defects when the training procedure as described above is followed.

Implementation details The ACGAN was trained on the input data in batches of size 25. The learning rate and β_1 were set at the same level as DCGAN, i.e., 0.0002 and 0.5 respectively. The discriminator of our ACGAN had 4 hidden layers with leaky ReLU activation function for 3 layers and linear activation function for 1 layer. The generator of the network also had 4 hidden layers with the ReLU activation function for all the layers. The ACGAN model was also trained for 100 epochs and 24 synthetic batches of the same batch size were generated for each class. Each batch was generated in the form of a grid of 5×5, which was sliced using the *image_slicer* library available in python. Once again, ACGAN was trained on a 5400 images-strong randomly selected subset (keeping classes balanced) of the classical augmentation training set, to generate 3600 samples. Three such training sets were generated and the mean score is reported in "Results and discussion" section.

InfoGAN

While changing the noise vector in GANs, it has consistent and meaningful effects on the generator output (Radford et al. 2015). There is no systematic way to find these structures because the generator uses the in a highly entangled way. Therefore, individual dimensions of z show no correspondence to semantic features in the data. The main idea of InfoGAN is to disentangle the representation space so as to decompose the domain knowledge existing in the data into a set of meaningful factors of variation (Chen et al. 2016).

The generator input is split into two parts: the noise vector and something referred to as the latent code vector. The latent code targets the structured semantic features in the data, made meaningful by maximizing the mutual information between the code and the generator output.

Since both the noise vector, z and latent code vector, c are provided to the generator, the form of the generator is now G(z,c), and the latent factors can be discovered in an unsupervised manner. Unlike in the original GAN, InfoGAN contains an information-theoretic regularization to handle the problem of the triviality of codes, i.e. the latent codes c and the distribution of the generator G(z,c) should have high mutual information (Fig. 6).

Overall, the image quality may not be as good as obtained from DCGAN. The major benefit of InfoGAN is that the dis-

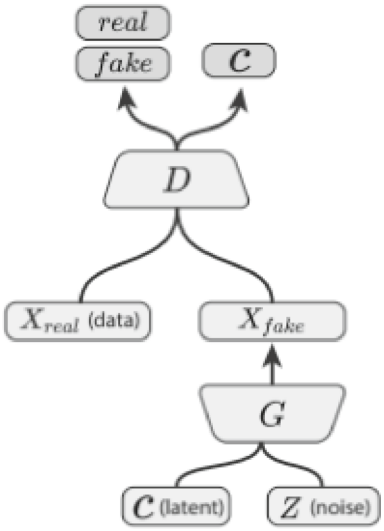


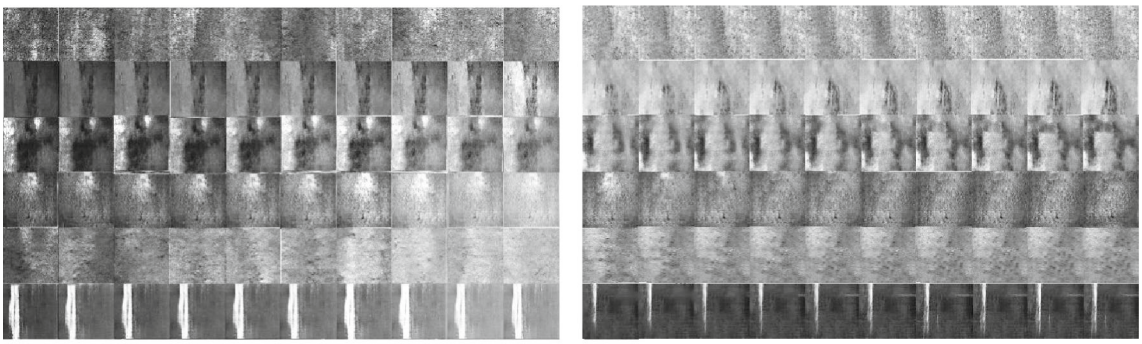
Fig. 6 Diagrammatic representation [figure taken from (Odena et al. (2017)] of the InfoGAN Architecture as per the original implementation given in Chen et al. (2016)

crete latent code allows for the conditional generation, and the continuous latent code allows for disentanglement of the latent space and control over the latent factors, as demonstrated in Fig. 7.

Architecture overview The architecture of the InfoGAN is effectively the same as the original GAN. However, it contains an extra regularization term over the original GAN's objective function. In practice, calculating mutual information is hard so standard variational arguments are used to approximate a lower bound. This involves introducing an auxiliary conditional distribution $Q(c|x)$, modeled by a parameterized neural network, and is meant to approximate the real conditional likelihood $P(c|x)$. Q and D as in the original GAN, share all convolutional layers, and one final fully connected layer output parameter for $Q(c|x)$.

InfoGAN contains a hyperparameter λ in the extra regularization term. It is easily tunable and set to 1 for discrete latent codes, whereas it's smaller is used for continuous latent codes. That ensures the L1 regularized term is on the same scale as the original GAN objectives. Since GANs are difficult to train, other approaches available on DCGAN in the literature are used to stabilize InfoGAN training (Chen et al. 2016).

Implementation details The InfoGAN was trained on the input data in batches of 30 images. The learning rate was set to 0.003 and, β_1 and β_2 were set to 0.5 and 0.999 respectively. The discriminator of the model had 3 hidden layers, with leaky ReLU as the activation function for all the layers. The generator had 5 hidden layers, out of which 4 had ReLU as an activation function while the final layer had a tanh activation function. The InfoGAN model was trained for 120 epochs, and synthetic batches of the same batch size were generated for each class. The grids were sliced using the *image_slicer* module available in python. In this experiment, two uniform continuous variables were used as latent codes. For the discrete latent code, the softmax nonlinearity was used, while for continuous latent codes it was found that simply treating the auxiliary distribution as a factored Gaussian was effective. The InfoGAN was also trained on a 5400 images-strong randomly selected subset (keeping classes balanced) of the classical augmentation training set, to generate 3600 samples, keeping classes balanced and randomly varying the continuous latent codes to synthetically



(a) *Latent code: Lighting and*

(b) *Latent code: Severity*

Fig. 7 **a** Latent code: Lighting and **b** Latent code: Severity. The discrete latent code ranging from 0 to 5 captures the 6 classes in the NEU-CLS dataset. The continuous latent codes vary from −1 to 1 and the effects are shown from left to right. In (**a**), the continuous latent code captures the variation in lighting; In (**b**), the categorical code controls the severity of the defect: for example, in the 2nd row which contains the inclusion class, the size of the affected area seems to grow from left to right. Similarly, in the 6th column, the scratched surface area also seems to grow from left to right. This indicates that the InfoGAN has learned to separate the size of the affected area by looking at a sufficient number of samples in the data distribution. This can help us control and add samples with additional domain knowledge into the augmented dataset

add additional domain knowledge (Fig. 7). Three such training sets were generated, and the mean score is reported in "Results and discussion" section.

Algorithm-GAN-based data augmentation

- Image data $x_1, x_2, x_3, \ldots, x_i$ with i images and their respective labels $y_1, y_2, y_3, \ldots, y_i$ were initially present. The total number of t classes was present. Therefore, $\{y_1, y_2, y_3, \ldots, y_i\} \in \{0,1,2,3, \ldots, i\}$.
- Build GANs for generating samples of known classes (in the case of DCGAN, t separate models for each class).
- Use this data to train a robust CNN classifier.
- Generate classically augmented images using the training data of j images per class.
- A subset of the resultant data with k images per class is used to train the GANs.
- Newly generated images $X_1, X_2, X_3, \ldots, X_k$ along with their corresponding class labels $Y_1, Y_2, Y_3, \ldots, Y_k$ are used to train the CNN.

Results and discussion

In the present study, an attempt has been made to compare the performance of CNNs for classification of surface defect data over the course of multiple experiments, where the CNNs are trained using different augmentation pipelines. An experiment is also performed with a pretrained CNN fine-tuned on the surface defect data, achieving state of the art results. In classification models, the confusion matrix is used for assessing the model performance.

All evaluation metrics for multi-class classification problems can be easily understood in the context of the binary classification problem (where the labels are either 0 or 1) when considering each class one-by-one. Basic terminology used in the confusion matrix is given as:

- *True Positive* (TP) Cases when the actual label is 1 and the predicted label is also 1, i.e., positive label was correctly predicted.
- *True Negative* (TN): Cases when the actual label is 0 and the predicted label is also 0, i.e., the negative label was correctly predicted.
- *False Positive* (FP): Cases when the actual label is 0 and the predicted label is 1. They are also known as 'Type I errors'.
- *False Negative* (FN): Cases when the actual label is 1 and the predicted label is 0. They are also known as 'Type II errors'.

The following terms are used to interpret the confusion matrix

- *Accuracy* It tells us how many labels were correctly predicted out of all the predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Sensitivity* It tells us how often, the model predicts the label as 1 when it is actually 1. It is also known as 'True positive rate' or 'Recall'.

$$Sensitivity = \frac{TP}{TP + FN}$$

- *Specificity* It tells us how often, the model predicts the label as 0 when it is actually 0. It is also known as 'True negative rate'

$$Specificity = \frac{TN}{TN + FP}$$

The proposed model performs multiclass classification of labels: RS, Pa, Cr, PS, In and Sc. The synthetically generated datasets were tested using simple CNN for the performance measures defined above. In the first experiment (Fig. 8), CNN was trained on dataset augmented using classic methods. The test accuracy was found out to be 90.28% for 20 epochs with a batch size of 60. Table 1 describes the confusion matrix for the first experiment. It can be observed from the confusion matrix that the model has good sensitivity for Cr, In, PS, and RS, but it is quite low for Pa and Sc. This results in a number of misclassifications for Pa and Sc. The specificity remains in an acceptable range for this model.

In the second experiment (Fig. 9), the CNN was trained for the DCGAN augmented dataset of the same size under similar training conditions, i.e., batch size of 60 for 20 epochs. The test accuracy was found out to be 95.78%, which is better for this dataset using a simple CNN without transfer learning. The sensitivity was within the acceptable range for all the classes, though it reduced insignificantly. However, the overall misclassifications were reduced by a significant amount. The specificity remained approximately the same for both the experiments.

In the third experiment (Fig. 10), CNN was trained for the ACGAN augmented dataset of the same size under the same training conditions and the test accuracy was found out to be 92.78%.

In the fourth experiment (Fig. 11), CNN was trained on the InfoGAN augmented dataset of the same size under the same training conditions. This experiment showed the second-highest test accuracy (lower than only the pipeline with DCGAN augmentation) of 94.86%. Despite the InfoGAN

Actual/Predicted	Cr	In	Pa	PS	RS	Sc	Sensitivity	Specificity
Cr	276	0	9	8	7	0	92.00%	99.07%
In	7	261	17	7	8	0	87.00%	96.07%
Pa	4	26	253	0	0	17	84.33%	97.27%
PS	3	0	1	296	0	0	98.67%	99.00%
RS	0	0	5	0	281	14	93.67%	99.00%
Sc	0	33	9	0	0	258	86.00%	97.93%

Fig. 8 Confusion matrix for experiment-1 (Classic Augmentation)

Table 1 Comparison of classic augmentation, DCGAN, ACGAN, and InfoGAN for CNN baseline

Augmentation method	Sensitivity		Specificity		Accuracy (%)
	Mean (%)	Std. Dev (%)	Mean (%)	Std. Dev (%)	
Classic augmentation	90.28	4.99	98.06	1.11	90.28
DC-GAN augmentation	95.33	3.03	99.16	0.37	95.78
AC-GAN augmentation	92.28	1.76	98.56	0.38	92.78
InfoGAN augmentation	94.06	2.78	98.81	0.36	94.86

Actual/Predicted	Cr	In	Pa	PS	RS	Sc	Sensitivity	Specificity
Cr	295	0	0	5	0	0	98.33%	99.20%
In	0	269	14	0	0	17	87.00%	99.20%
Pa	4	5	284	0	7	0	94.67%	98.87%
PS	3	0	3	289	1	4	96.33%	99.67%
RS	4	4	0	0	291	1	97.00%	99.47%
Sc	1	3	0	0	0	296	98.67%	98.53%

Fig. 9 Confusion matrix for experiment-2 (DCGAN Augmentation)

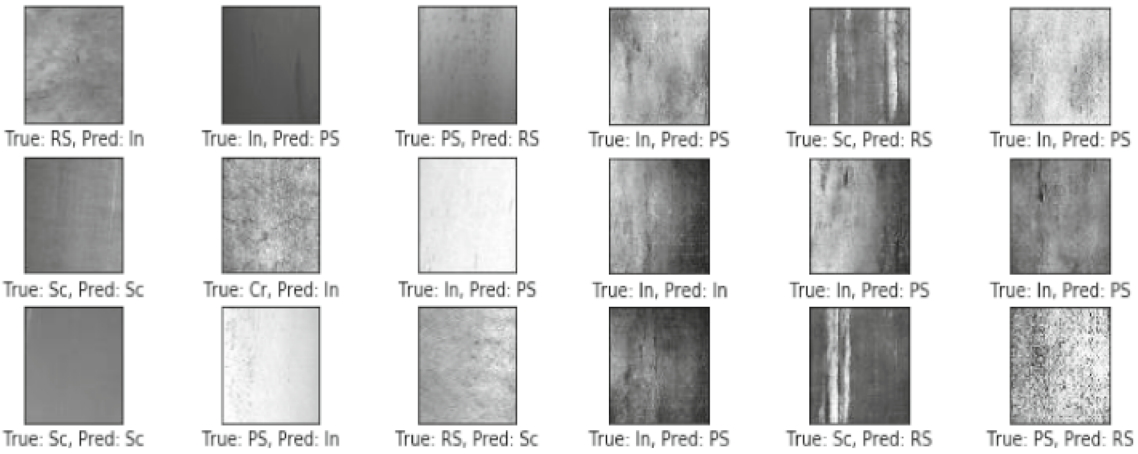
Actual/Predicted	Cr	In	Pa	PS	RS	Sc	Sensitivity	Specificity
Cr	280	0	0	11	9	0	93.33%	98.33%
In	0	270	14	0	0	16	87.00%	98.33%
Pa	7	9	273	0	11	0	91.00%	98.80%
PS	3	0	4	283	3	7	94.33%	99.27%
RS	9	7	0	0	279	5	93.00%	98.47%
Sc	6	9	0	0	0	285	95.00%	98.13%

Fig. 10 Confusion matrix for experiment-3 (ACGAN Augmentation)

Actual/Predicted	Cr	In	Pa	PS	RS	Sc	Sensitivity	Specificity
Cr	289	0	0	8	3	0	96.33%	98.53%
In	0	271	10	0	0	19	90.33%	98.67%
Pa	2	10	279	0	9	0	93.00%	99.13%
PS	11	0	3	273	9	4	91.00%	99.47%
RS	9	3	0	0	288	0	96.00%	98.60%
Sc	0	7	0	0	0	293	97.67%	98.47%

Fig. 11 Confusion matrix for experiment-4 (InfoGAN Aug.)

augmented dataset containing additional domain information, the accuracy was lesser than that of the DCGAN augmented pipeline. This is due to the image fidelity being lesser.



(a) *Classic augmentation* **(b)** *DC-GAN augmentation*

Fig. 12 Predicted examples

Table 2 Comparison of the feature extraction approaches for classification, namely AECLBP, SCN, Overfeat, DCGAN augmented CNN baseline, and DCGAN augmented finetuned CNN

Classification method	Accuracy (%)
AECLBP	97.22
SCN	98.60
Overfeat	98.70
CNN Baseline	95.78
Finetuned CNN	99.11

Figure 12 shows predicted examples from classic augmented model and DC-GAN augmented model.

Table 2 analyses the results of different augmentation methods used in our study. Training the dataset using the DC-GAN augmentation method, yielded better results compared to the classic augmentation method as well as AC-GAN and InfoGAN augmentation methods. The CNN trained using DC-GAN augmentation has a better sensitivity than the other two methods, which signifies that it can predict positive labels more accurately. It also has higher specificity which implies that the CNN trained using the DC-GAN augmentation method has a higher negative label prediction ability. It also has higher overall accuracy as a result of its higher overall classification ability.

From Table 1 it can be inferred that DCGAN Augmentation gives the best results. Therefore, in the fifth experiment (Fig. 13), the pretrained CNN, described in "Convolutional neural network" section, was fine tuned on the training data. The training conditions were kept the same and the test accuracy achieved was 99.11%.

In the final experiment (Fig. 13) the classification result for feature extraction from our proposed augmentation scheme was compared with the several other methods, surpassing them with the fine tuned CNN. Song and Yan's approach (Song and Yan 2013) introduces and uses the Adjacent Evaluation Completed Local Binary Pattern (AECLBP) feature descriptor for the SVM. Feature extraction through Scattered Convolutional Network (SCN) (Song et al. 2014), and Overfeat Network (Sermanet et al. 2014) for classification are also compared. Our approach simply utilizes the features extracted from CNN. The impact of an information-rich training set generated from our GAN based augmentation scheme is that those features are enough for a higher classification performance. So, our approach has the added advantage of not requiring any feature-engineering. That is why a simple CNN trained on synthetically augmented data outperforms more complex CNN based feature extractors trained with classic augmentation.

Conclusion

The quality of the product plays an important role in determining the effectiveness of a manufacturing operation. In the manufacturing industry, inspection not only plays an important role in quality control, but it also works as a feedback system for the process. Hence, defect classification is as important as defect detection.

In this study, GANs are trained to augment the data for surface defects on hot-rolled steel strips. The framework has been developed in order to investigate the use of synthetic augmentation for classification of manufacturing datasets. Thus, a synthetic data augmentation scheme is developed and compared with classic data augmentation over the course of five experiments. Initially, the training set was enlarged to a size of 9000 images using graphical transformation methods such as flipping, rotation, and scaling. The validation accuracy of the CNN model was found out

Actual/Predicted	Cr	In	Pa	PS	RS	Sc	Sensitivity	Specificity
Cr	300	0	0	0	0	0	100%	99.60%
In	1	296	3	0	0	0	98.67%	99.60%
Pa	1	3	295	0	1	0	98.33%	99.80%
PS	0	3	0	297	0	0	99.00%	100%
RS	1	0	0	0	299	0	99.67%	99.93%
Sc	3	0	0	0	0	297	99.00%	100%

Fig. 13 Confusion Matrix for Experiment-5

to be 90.28%. In the second experiment, the CNN is trained using DCGAN-augmented data having an identical size and under the same training parameters. The models predicted the defect classes with an accuracy of 95.78%, which was a significant improvement over the model proposed in the first experiment. In the third experiment, DC-GAN is substituted with AC-GAN, keeping the remainder of the data augmentation pipeline identical. The accuracy of this model was found out to be 92.78%. The accuracy of AC-GAN was observed to be less than that of DC-GAN by 3%. In the fourth experiment, InfoGAN is employed, keeping the remainder of the data augmentation pipeline identical and the accuracy of this model was found to be 94.86%, less than that of DC-GAN by 0.92%.

Overall, for hot-rolled steel strips, an improvement in the performance of 5.5% is achieved using DCGAN-based augmentation over Classic augmentation. The observed experimental results show the efficacy and robustness of the proposed model and can be used as a benchmark. In our fifth experiment, using a pretrained CNN for transfer learning, state of the art results on NEU CLS were achieved obtaining a performance gain of 3.33% over the baseline.

The main finding of this work is that the proposed method shows great potential for being adopted as a standard augmentation technique for deep learning for industrial applications, which generally have small-sized datasets. The suggested GAN-based data augmentation procedure offers a great advantage by significantly cutting down the time taken for, and the opportunity cost of, collection of real-world data. Furthermore, our pipeline provides another benefit—it may be used to train successful models for other applications (such as for structural defects using X-ray images) through domain adaptation, reducing the time even further. It is also easy to run (once trained) and thus can be physically employed at a manufacturing plant using cameras and single-board computers.

This study can be extended to other models. As the classification is performed using a basic CNN, any development on the base architecture will also be reflected in the final model. This has been demonstrated using transfer learning. Similarly, more advanced GAN models may also positively influence the augmentation process.

Future research could scrutinize the segmentation problem for manufacturing defect datasets. This is an inspiring problem as it is required to generate both synthetic surface defect images and their ground truth masks. The applied methodology may be extended for the purpose to improve industry performance in different sectors.

References

Antoniou, A., Storkey, A., & Edwards, H. (2018). Augmenting image classifiers using data augmentation generative adversarial networks. In *Artificial neural networks and machine learning—ICANN 2018*.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN*.

Badmos, O., Kopp, A., Bernthaler, T., & Schneider, G. (2020). Image-based defect detection in lithium-ion battery electrode using convolutional neural networks. *Journal of Intelligent Manufacturing, 31,* 885–897.

Berthelot, D., Schumm, T., & Metz, L. (2017). *BEGAN: Boundary equilibrium generative adversarial networks*.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., & Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Carreira-Perpiñán, M. Á., & Hinton, G. E. (2005). On contrastive divergence learning. In *AISTATS*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*.

Davtalab, O., Kazemian, A., Yuan, X., & Khoshnevis, B. (2020). Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection. *Journal of Intelligent Manufacturing*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, Miami, FL, USA.

Devadas, C., Samarasekera, I. V., & Hawbolt, E. B. (1991). The thermal and metallurgical state of steel strip during hot rolling: Part III. Microstructural evolution. *Metallurgical Transactions A, 22*(2), 335–349.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., et al. (2015). Flownet: Learning optical flow with convolutional networks. In *ICCV*.

Feng, S., Zhou, H., & Dong, H. (2019). Using deep neural network with small dataset to predict material defects. *Materials and Design, 162,* 300–310.

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 (ISBI 2018)*, Washington, DC.
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative Adversarial Nets. In *Advances in neural information processing systems 27*.
Grzenda, M., & Bustillo, A. (2019). Semi-supervised roughness prediction with partly unlabeled vibration data streams. *Journal of Intelligent Manufacturing, 30,* 933–945.
Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). *Improved training of Wasserstein GANs*.
Hao, R., Lu, B., Cheng, Y., Li, X., & Huang, B. (2020). A steel surface defect inspection approach towards smart industrial monitoring. *Journal of Intelligent Manufacturing*.
Hinton, G. E., Dayan, P., & Frey, B. J. (1995). The "Wake-Sleep" Algorithm for Unsupervised Neural Networks. *Science, 268,* 1158–1161.
Hjelm, R. D., Jacob, A. P., Che, T., Trischler, A., Cho, K., & Bengio, Y. (2018). Boundary-seeking generative adversarial networks. In *ICLR*.
Huang, Y., Qiu, C., Guo, Y., Wang, X., & Yuan, K. (2018). Surface defect saliency of magnetic tile. In *IEEE international conference on automation and engineering*, At Munich, Germany.
Izadi, S., Mirikharaji, Z., Kawahara, J., & Hamarneh, G. (2018). Generative adversarial networks to segment skin lesions. In *IEEE 15th international symposium on biomedical imaging*, Washington, DC, USA.
Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. In *ICLR*.
Lai, Y. T. K., Hu, J. S., Tsai, Y. H., & Chiu, W. Y. (2018). Industrial anomaly detection and one-class classification using generative adversarial networks. In *IEEE/ASME international conference on advanced intelligent mechatronics (AIM)*.
Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., & Yan, S. (2017). Perceptual generative adversarial networks for small object detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, USA.
Luo, J., Huang, J., & Li, H. (2020). A case study of conditional deep convolutional generative adversarial networks in machine fault diagnosis. *Journal of Intelligent Manufacturing*.
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, vol. 10574*.
Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Smolley, S. P. (2017). Least squares generative adversarial networks. In *IEEE international conference on computer vision,* Venice, Italy.
Mirza, M., & Osindero, S. (2014). *Conditional generative adversarial nets*.
Moeskops, P., Veta, M., Lafarge, M. W., Eppenhof, K. A., & Pluim, J. P. (2017). Adversarial training and dilated convolutions for brain MRI segmentation. In *Workshop on deep learning in medical image analysis*.
, V., Mohammed Safwan, K. P., Chennamsetty, S. S., & Krishnamurthi, G. (2017). Generative adversarial networks for brain lesion detection. In *SPIE medical imaging*, Orlando, Florida, United States.
Odena, A., Olah, C., & Shlens, J. (2017). *Conditional image synthesis with auxiliary classifier GANs*.
Pan, J., Canton, C., McGuinness, K., O'Connor, N., Torres, J., Sayrol, E., et al. (2017). SalGAN: Visual saliency prediction with adversarial networks. *Computer Vision and Image Understanding*.
Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*.
Richter, S.R., Vineet, V., Roth, S., & Koltun, V. (2016). Playing for data: Ground truth from computer games. In *ECCV*.
Scime, L., & Beuth, J. (2018). Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing, 19,* 114–126.
Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International conference on learning representations*, Banff.
Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., & Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *CVPR*.
Song, K., Hu, S., & Yan, Y. (2014). Automatic recognition of surface defects on hot-rolled. *Journal of Computational Information Systems, 10*(7), 3049–3055.
Song, K., & Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science, 285,* 858–864.
Song, K., & Yan, Y. (2019). NEU surface defect database. Northeastern University. http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html. [Accessed 5 4 2019].
Sun, T., Tien, F., Tien, F., Tien, F. C., & Kuo, R. J. (2016). Automated thermal fuse inspection using machine vision and artificial neural networks. *Journal of Intelligent Manufacturing, 27,* 639–651.
Tabernik, D., Šela, S., Skvarč, J., & Skočaj, D. (2020). Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing, 31,* 759–776.
Tian, Y. (2017, April 16). Master Chinese calligraphy with conditional adversarial networks. https://kaonashi-tyc.github.io/2017/04/06/zi2zi.html. [Accessed 10 5 2019].
Wolterink, J. M., Leiner, T., Viergever, M. A., & Išgum, I. (2017). Generative Adversarial Networks for Noise Reduction in Low-Dose CT. *IEEE Transactions on Medical Imaging, 36*(12), 2536–2545.
Yang, D., Xu, D., Zhou, S. K., Georgescu, B., Chen, M., Grbic, S., et al. (2017) Automatic liver segmentation using an adversarial image-to-image network. In *International conference on medical image computing and computer-assisted intervention*, Quebec City, QC, Canada.
Yu, H., Tieu, K., Lu, C., Deng, G., & Liu, X. (2013). Occurrence of surface defects on strips during hot rolling. *International Journal of Advanced Manufacturing Technology, 67,* 1161–1170.
Zhai, W., Zhu, J., Cao, Y., & Wang, Z. (2018). A generative adversarial network based framework for unsupervised visual surface inspection. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Calgary, AB, Canada.
Zhang, Z., Song, Y., & Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
Zhao, J., Mathieu, M., & LeCun, Y. (2017). Energy-based generative adversarial networks. In *ICLR*.
Zhu, X., Liu, Y., Li, J., Wan, T., & Qin, Z. (2018). Emotion classification with data augmentation using generative adversarial networks. In *Advances in knowledge discovery and data mining. PAKDD 2018*, 2017.