Line-Post Insulator Fault Classification Model using Deep Convolutional GAN-based Synthetic Images

Dongjoo Kim, *Student Member, IEEE*, Subir Majumder, *Member, IEEE*, and Le Xie, *Fellow, IEEE*

***Abstract*—Due to thermal, electrical, mechanical, and chemical stresses, line-post insulators in the power system may degrade over time. The degradation process continuously gets exacerbated by the above-mentioned factors. Therefore, condition monitoring of line insulators must be frequently carried out. Optical cameras are considered the most accurate among existing technologies for detecting such defects. Computer vision techniques aided by optical cameras could automate faulty insulator identification. However, there is a limited size of the training data set obtained from real-world optical camera images. In this paper, we propose a generative approach to creating a massive amount of line-post insulator fault images through Deep Convolutional Generative Adversarial Networks (DCGAN). The additional training data obtained from DCGAN-based approach is shown to improve the accuracy of the insulator fault classification. In the case study, we show that with an increasing number of synthetic images created by DCGAN, the accuracy of the fault classification continuously improves. The ability to classify true faulty insulators has increased from 56% to 94%. The performance of the DCGAN-based approach is also compared with the random oversampling approach. The numerical results suggest that the DCGAN-based approach has the advantage of detection accuracy and a lower false positive rate.**

***Index Terms*—Generative adversarial networks (GAN), Insulator fault diagnostic, Synthetic data augmentation**

I. Introduction

Line-post insulators are specifically engineered to insulate transmission and distribution lines from the supporting structures, such as utility poles or towers, which are typically grounded. These insulators are constructed from materials that exhibit high dielectric strength, such as ceramic or glass and are designed to withstand substantial electrical stresses. However, these insulators are exposed to the environment and constantly subjected to thermal, electrical, mechanical, and chemical stresses, which can create degradation spots (such as pin holes or cracks). Subsequently, depending on the natural environment, these degradation spots expand (due to salt or moisture introduction or electrical treeing following sustained electrical stresses). Damage to these insulators can trigger arcing or ground leakage, which is hazardous to civilians and maintenance crews. Therefore, it is extremely important to accurately identify the defective line-post insulators in advance for the reliable operation of the power grid.

Multiple methods, including optical diagnostics [1], ultrasonic testing, infrared thermography [2], and leakage current measurement exist for the identification of equipment failure within the power infrastructure facilities. For a typical power system such as the Korean power grid, we illustrate the state-of-the-art performance of different inspection methods in terms of their identification accuracy of faulty components. As shown in Table I (see, [3] for details), optical diagnostics, specifically visual inspection, provides the highest diagnostic yield for faulty equipment classifications including line-post insulators and has been widely used in the field [4]. Another observation from this table is only $\sim 5\%$ faulty equipment images are available from a pool of $\sim 1.5M$ diagnostic images taken. While collecting such a large amount of optical images and manual diagnosis could be physically demanding, the use of unmanned aerial vehicles (UAV) [5], such as drones, could simplify the image collection process. Recent advances in computer vision technologies along with the use of UAVs could largely simplify power system equipment failure diagnostics, but, limited available images of faulty insulators create a severe bottleneck in the model development for the classification of faulty line-post insulators.

Classification is a sub-category under supervised learning algorithms, where the classifier is provided with labeled data sets to understand the latent features. There are multiple challenges when there is skewness in the available data set [6], some of them are: (i) the classifier may not be able to generalize patterns within certain modes, (ii) it may be difficult to accurately identify the decision boundary. One of the traditional approaches for reducing the skewness in the available data set is to use oversampling techniques [7]. This method involves duplicating samples from the minority class (for example, random duplication as discussed in [8]), and reducing the number of samples drawn from majority classes. While this method artificially increases the training samples, its major drawback is that the oversampling can lead to overfitting, particularly when the generated data closely resembles the original data. Another alternative contrary to this approach could be the use of generative approaches, such as the Generative Adversarial Network (GAN) [9]. Instead of duplicating existing samples, GAN can generate diverse samples from scratch, tremendously improving the performance of artificial intelligence-based classifiers.

Given the described problem of identifying faulted line-post insulators based on images, one needs to use convolutional layers to understand spatial dependencies, which can be one of the reasons behind the use of Deep Convolutional Generative Adversarial Network (DCGAN) [10] for augmenting existing image database with synthetic images of defective insulators. The contribution of this paper is, therefore, two-fold:

i. This paper demonstrates how GAN-based approaches could improve the quality of data classification in the context of equipment degradation. As a use case, we focus on classifying faulted line-post insulators.
ii. This paper introduces a DCGAN-based approach to creating synthetic images of line post-insulator faults and compares the impact of changing data-set sample sizes.

The rest of the paper is organized as follows. Section II, provides a rough overview of DCGAN and how it could be utilized for generating realistic faulty images of line-post insulators. Section III describes the ResNet-based classification model and some parametric analysis to demonstrate how synthetic images are assisting the classifier. A comparison

Dongjoo Kim, Subir Majumder, and Le Xie are with the Department of Electrical & Computer Engineering, Texas A&M University, College Station, Texas 77843. (e-mail: le.xie@tamu.edu).

This work is partially supported by the Texas A&M Engineering Smart Grid Center and Texas A&M Energy Institute.

TABLE I
IDENTIFYING EQUIPMENT FAILURE STATISTICS THROUGH DIVERSE METHODOLOGIES IN ACTUAL POWER INFRASTRUCTURE FACILITIES

Power facilities / Inspection methods	Total number of diagnosis	Classified faulty equipments					Total number of fault	Faulty Equipment Detection Success Rate [%]
		Insulator	Cut out switch / Lightning arresters	Transformer	Switch	Others		
Thermal image	5,249,659	5,792	11,953	504	593	9,404	28,246	0.54
Ultrasonic	3.776,685	10,605	4,313	335	1,058	4,506	20,817	0.55
Optical	1,648,544	38,523	35,351	1,547	596	18,087	94,104	5.71
Others	452,007	969	685	9	1,009	495	3,167	0.70

with random oversampling-based data augmentation is also presented. Section IV concludes this paper.

II. Uses of GAN in Generating Synthetic Images

In this section, we present the problem formulation of GAN-based synthetic image creation for line-post faults.

A. Generative Adversarial Networks (GAN) Overview

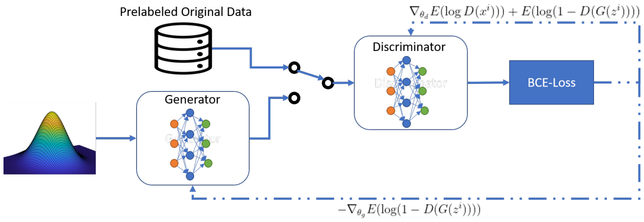

Prelabeled Original Data
$\nabla_{\theta_d} E(\log D(x^i)) + E(\log(1 - D(G(z^i))))$
Discriminator
BCE-Loss
Generator
$-\nabla_{\theta_g} E(\log(1 - D(G(z^i))))$

Fig. 1. Synthetic data generation using GAN

The GAN framework aims to learn the distribution of latent variables in the example data set and try to capture such distribution in the generated data. A detailed overview of the GAN-based architectures is described in [11], [12]. As discussed in the existing literature, GAN consists of a generator, G, and a discriminator, D, both of which are typically neural networks based on multi-layer perceptrons (MLPs). As shown in Fig. 1, the generator model aims to develop realistic data sets from random noises, and the discriminator tends to classify images into real and synthetic ones. Here, as shown in (1), the generator and the discriminator play a min-max game, which is based on binary cross entropy (BCE) loss. Suppose p_z is the noise distribution, then the generator, $G(z, \theta_g)$, a differentiable function, maps noise space (or latent vector) (z) to data space with the parameter, θ_g. The discriminator, $D(x, \theta_d)$, is a scalar with the model parameter θ_d, which calculates the probability with which the input samples (x) are from the original database. Therefore, the generator is aimed to minimize the expected log of $(1 - D(G(z))$, which calculates how far ahead are the probability distributions of true and generated data. The discriminator acts as a classifier to maximize that both real and synthetic samples are assigned correct labels[1]. Consequently, both generators and discriminators receive the gradient of the associated loss function for learning the underlying distribution. The learning completes at a saddle point where the objective function, $V(D, G)$, reaches a maximum with respect to the discriminator and the minima with respect to the discriminator — a Nash equilibrium.

$$\begin{aligned}\min_G \max_D V(D, G) &= E_{x \sim p_{\text{data}}}[\log D(x)] \\ &+ E_{z \sim p_z}[\log(1 - D(G(z)))]\end{aligned} \tag{1}$$

Therefore, GANs aim to capture the hidden distribution of latent variables within the input data and their nonlinear, high-dimensional cross-dependencies. Non-linear activation functions within the generator and discriminator MLPs help encapsulate these hidden non-linearities. The non-linear property of the BCE-loss function is able to emphasize misclassification. Given the underlying non-convex optimization problem, GANs do not always have a unique solution, and based on the chosen different initialization and hyperparameters, the solution may get stuck at a local optimum. Furthermore, the optimal solution of a GAN corresponds to the saddle point. Secondly, the use of gradients in learning creates a condition, especially during the initial part of the training, where the generator fails to capture the distribution of latent variables, and the discriminator marks all the generated data set as synthesized. Or, the generator produces a similar set of data, which is widely known as mode collapse.

B. Using DCGAN for Generating Artificial Images

Despite these challenges, GANs, being a powerful generative model, has found widespread application in computer vision and image processing, generating sequential data set, etc. Contrary to typical data sets, image data (in the current context) contains spatial relationships and convolutional operation helps in capturing these relationships [13]. Typically, convolutional neural networks (CNNs) consist of a set of filters, also known as kernels or feature detectors, that slide over the input image to perform convolutions and extract relevant features (refer to [13] for a detailed treatment). The length by which the kernel slides is known as the stride length. In Fig. 2, the input image is of size 5×5, the kernel is of size 3×3, and the stride length is 1. Following the convolution layer, the CNN also includes an activation function layer (introducing nonlinearity into the network) and a pooling layer (a downsampling operation that reduces the dimensionality of the feature map). Therefore, the convolution and activation part of the neural network is carried out using neural networks, and the objective would be to learn these kernels for better extraction of features. Once the features are extracted, these feature sets are passed through a fully connected layer for classification.

A CNN consists of multiple of these kernels/filter layers, aiming to capture all of the complex feature sets, and the resulting output feature map preserves spatial relationships. A typical CNN architecture is given in Fig. 3:

Consequently, the majority of the GANs that are developed to work on image data, such as Conditional GAN, DCGAN, and CycleGAN, utilize a convolutional layer because of their effectiveness in understanding spatial dependencies and capturing local patterns. Here, our focus will be on the use of DCGAN for generating synthetic images. Originally introduced in [9], DCGAN is an extension of the original GAN architecture, where the generator and discriminator are modified with convolutional layers (see Fig. 4), where, as

[1] GAN does not require the input data set to be labeled.

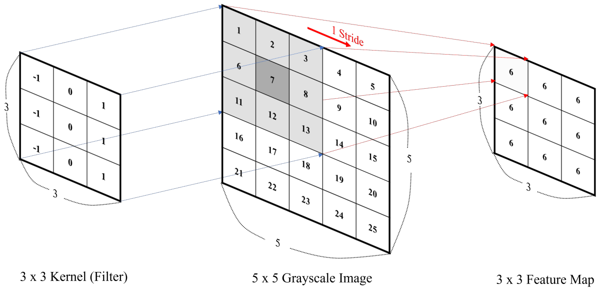


Fig. 2. Sample convolutional layer for a typical gray-scale image using a 3×3 kernal

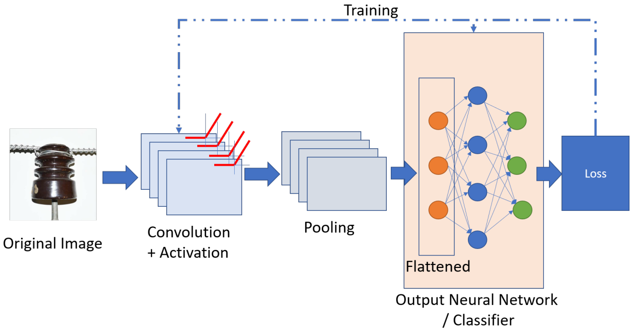


Fig. 3. Typical CNN design

discussed, the kernels of both generators and discriminators are required to be trained suitably. One of the major issues of using CNN in image classification is the use of a fully connected layer in the output, where, given the spatial relationship, the researchers often argue for the use of partially connected layers. Secondly, while the ReLU activation function is typically used in CNN, it becomes inactive when the inputs become negative, significantly affecting the learning during the training process.

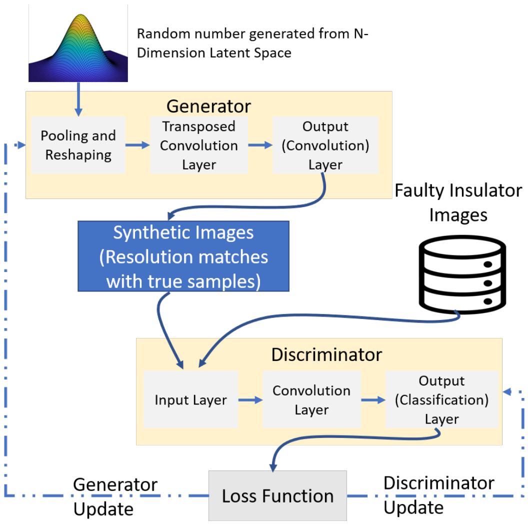

Random number generated from N-Dimension Latent Space
Generator
Pooling and Reshaping
Transposed Convolution Layer
Output (Convolution) Layer
Faulty Insulator Images
Synthetic Images (Resolution matches with true samples)
Discriminator
Input Layer
Convolution Layer
Output (Classification) Layer
Generator Update
Loss Function
Discriminator Update

Fig. 4. Generating synthetic defected line post insulator with DCGAN

To circumvent the same, DCGAN uses the leakey ReLu activation function in its convolution layer (see the comparison in Fig. 5). Thirdly, pooling in the convolution layer is replaced by strided convolution (stride length greater than 1). DCGAN uses other techniques, such as batch normalization, which increases the stability of the training process quite significantly, as well as partially connected layers.

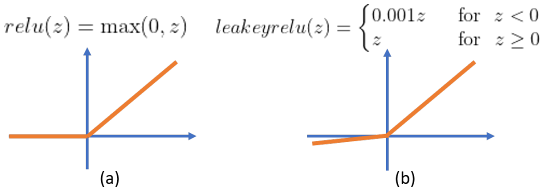


Fig. 5. ReLU vs. Leakey ReLU: Architectural Difference

Like other GAN frameworks, DCGAN also has a generator and a discriminator, which we will discuss in detail in the following subsection considering line-post insulator fault diagnostic as a use-case.

C. *Generating Synthetic Image using DCGAN*

The training process of DCGAN typically consists of the following steps:

1) Data preparation: Although the input images in CNNs are not typically downsampled, they are downsampled as a part of DCGAN. This is to ensure a balance between computational efficiency, spatial resolution, and information preservation. Here, all of the obtained images will be downsized to 64 × 64. Fig. 6 depicts some of the actual images of a damaged line-post insulator. Given the objective of the GAN-based generator is to generate images of faulty insulators, in this research, 52 pre-labeled images of defective line-post insulators were used for training the GAN.

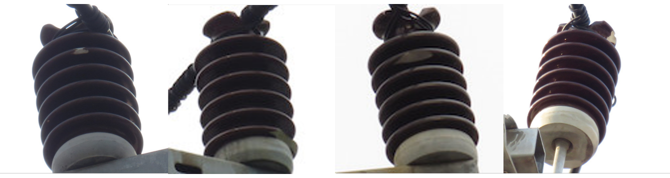


Fig. 6. Real images of defected line post insulator

2) Generator and discriminator initialization: Like the GAN framework, in the current context, DCGAN strives to map randomly generated vectors into an image set consisting of line-post insulators based on certain parameters θ_g (see the overview on GAN, and Fig. 4 for the overall architecture of DCGAN). Given the focuses are on images, we seek the help of transposed convolution to map the random numbers into the images. First, the random noise vector is projected through a fully connected layer with learnable weights and reshaped into a tensor of suitable dimensions and channels. The overall process helps in applying transposed convolution[2] sequentially for upscaling the image. In this work, the size of the latent vector (z) is chosen to be 4[3]. DCGAN uses strided convolution instead of pooling and batch normalization to stabilize the training process. Leakey ReLU activation function is used to capture the nonlinearity within the images while minimizing

[2] Transposed convolution operation is very similar to convolution operation, where both involve input data and a learnable kernel. However, as opposed to feature extraction, here, the objective is building the image based on features.

[3] Typically, this number should be more than one and less than the number of test samples for training. In this problem, we have treated this variable as a hyperparameter.

the impacts, such as dead neurons and the ‘tanh’ activation layer in the output to produce the synthesized image.

The discriminator is a typical CNN, designed to take 64×64 RGB-image as an input. The discriminator also utilizes strided convolution instead of pooling and batch normalization to improve stability. It utilizes the Leakey ReLU activation function with a ‘sigmoid’-based output layer. All the weights in the generator and discriminators are suitably initialized.

3) Adversarial training: DCGAN is trained in a process where the generator and discriminator are trained together. During training, the discriminator is trained on a batch of real images from the data set and a batch of generated images from the generator. The generator is a more complex process, given that it has to learn all latent information within the real samples. The training process in the DCGAN is classified into epochs and iterations. The epochs refer to generators producing a batch of synthetic images. Ideally, at the beginning of each epoch, the generator produces synthetic images, and the discriminator is trained to identify both synthetic and real images for each mini-batches. Within a given epoch, neural networks within generators and discriminators are updated multiple times based on the determined gradient. Here, the number of training epochs was 100. The objective is to minimize the generator’s loss and improve its ability to generate more realistic images. As the training progresses, the generator learns to generate more realistic images, while the discriminator becomes more accurate in distinguishing real and synthetic images.

D. Synthetic results

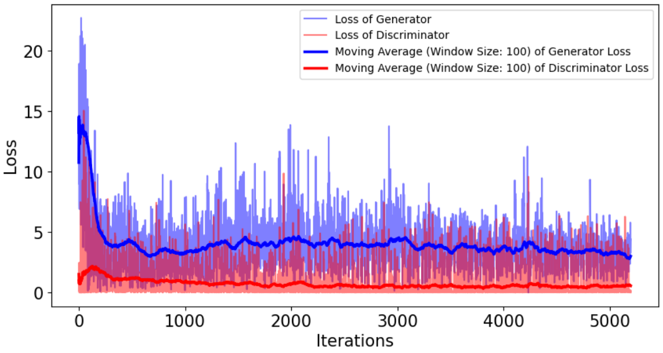


Fig. 7. Generator and discriminator loss function across iterative training

As the training progresses, the model learns to generate images that exhibit characteristics similar to real images, and the discriminator becomes more adept at differentiating between real and synthetic images. Fig. 7 shows a loss of generator and discriminator during training. In the initial stages of training, both the generator and discriminator losses are usually high, as the model starts with random weights and biases. The trend of the losses varies during the training process. Initially, the discriminator loss decreases faster than the generator loss as the discriminator becomes more accurate at distinguishing between real and synthetic images. This can result in the generator struggling to generate realistic images and experiencing a higher loss. As the training continues, the generator starts to improve its ability to synthesize more realistic images, this leads to a decrease in the discriminator loss. As the training progresses, the losses for the generator and discriminator should converge to a relatively stable state.

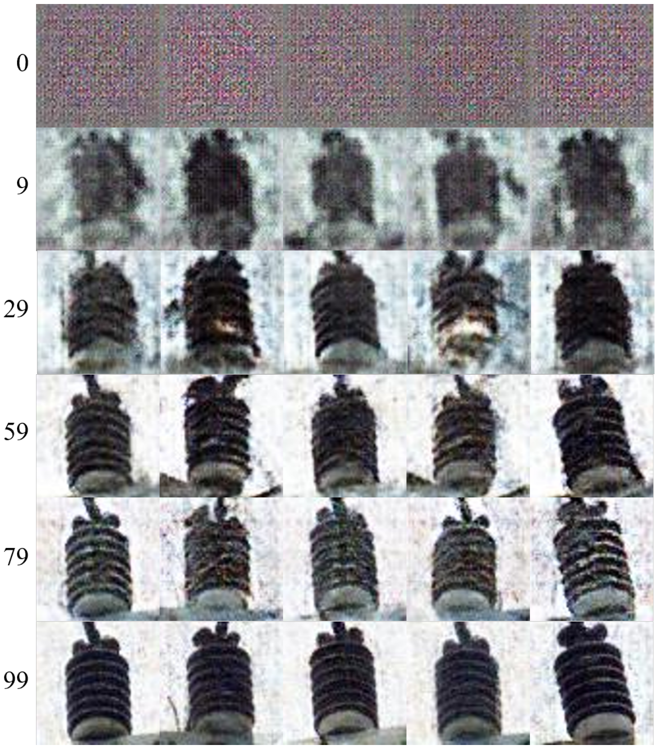


Fig. 8. Evolution of Synthetic images of defected line post insulator generated by DCGAN across multiple epochs

The evolution of synthetically generated faulty insulator images across multiple epochs is demonstrated in Fig. 8. Due to the complex nature of image generation, the generated images after the final epoch may still exhibit some imperfections, commonly referred to as “noise,” in the background or other regions. The generated images from the designed DCGAN can also have a grainy background, but the images so synthesized can still be considered relatively valid, as they demonstrate a resemblance to real images. The noise in the background can be interpreted as a result of the model’s attempt to learn and reproduce the intricate details and patterns present in real images. Note that, the discriminator being a classifier, it learns to selectively ignore the ‘graininess’ and becomes better and better at identifying synthetic images. One possible way to tackle this challenge could be to add additional noise with the true images for training.

III. An Enhanced Classification Model using Synthetic Images: A Case Study

The effectiveness of the synthetic images in the line-post insulator fault diagnostic model — enabling true power system automation through computer vision — is demonstrated in this section. Here, we have used ResNet [14] model, a CNN, to provide us with the desired computer vision. In this section, first, we provide a brief introduction on ResNet. Secondly, a description of the utilized data set and the metrics utilized for understanding model performances are elaborated. Subsequently, we will provide a comparative analysis of model performance through the case studies:

Case A: How the model learns to identify the false positives to true negatives with the addition of more synthetic images using DCGAN, mitigating the skewness of the ResNet training data.

Case B: How the model learns to identify the false positives to true negatives with the addition of more synthetic images using DCGAN, when there is no skewness in the ResNet training data, and compare the performance of the added data with an example oversampling technique (such as random oversampling [8]).

A. ResNet: An Image Classification Model

Given that deep neural networks consist of multiple layers, each of which is required to be updated based on the calculated gradient, they suffer from issues such as vanishing gradient, where the gradient of the loss function becomes close to zero and the neural network cannot update itself. The architectural change introduced by ResNet compared to a typical CNN model is that instead of directly learning the mapping of the relationships among the images and objects, it uses a residual layer (output of one layer is taken and subsequently added to the output of another layer ahead — also known as the skip connection. See Fig. 9). ResNet comes with different architectures based on the number of layers, however, given the effectiveness, and computational efficiency, we have decided to use ResNet-18 for classifying the condition of line-post insulators. We have used cross entropy as a loss function and a Stochastic Gradient Descent (SGD) technique for training the neural network.

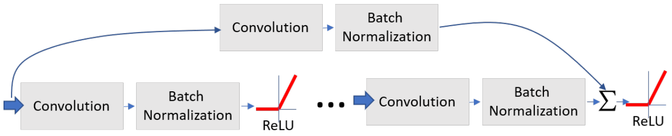


Fig. 9. Residual layer in ResNet.

B. Data sets, Parameter Setting and Comparison Metric

Testing Data: Our data set consists of 52 damaged pre-labeled line-post insulator images and 223 normal images from the field. Based on a typical 7:3 ratio, we have separated out 16 damaged and 66 undamaged insulators beforehand for testing. We will be using the same data sets across all the case-study to ensure comparability of the results across multiple models (note that we will not be utilizing synthetic images for testing). In the training process, the learning rate is 0.001, and the number of epochs is 6.

Training Data: Five sub-cases are considered as a part of Case A, where multiple images of synthetically generated damaged insulators were added to the data set, resulting in different skewness ratios as presented in Table II. Here we ensured that the synthetic images were retained across cases, i.e., for data set 3, we reused 27 synthesized images from data set 2, and 37 DCGAN-generated images were also added, making a total of 64 synthetic images as a part of the data set 3. Such a treatment would allow us to specifically see the impact of added synthetic images.

Five sub-cases are considered as a part of Case B, where we randomly selected 36 samples (without repetition) from 156 normal insulator images. In the subsequent sub-cases, for the faulty insulators, we have selected DCGAN-generated synthetic images, or, based on random over-sampling. For the normal insulator images, we have added new images from the real database while ensuring that the skewness always remains 1:1.

TABLE II
DATA SET USED FOR CLASSIFICATION WITH RESNET

	Normal	Fault (Actual + Synthetic)	Skewness Ratio
Data set 1	156	36 + 0	1:0.2
Data set 2	156	36 + 27	1:0.4
Data set 3	156	36 + 64	1:0.6
Data set 4	156	36 + 91	1:0.8
Data set 5	156	36 + 120	1:1

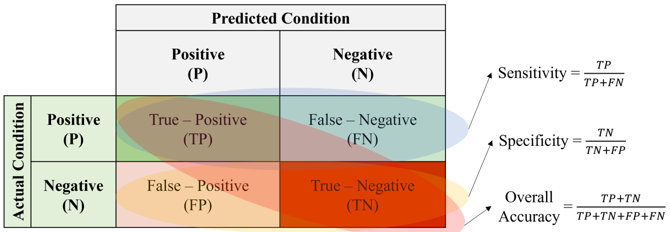


Fig. 10. Metric to compare the overall performance of ResNet models

Comparison Metric: The performance of the classification model was evaluated using the confusion matrix and the receiver operating characteristics (ROC) curves. Typically four possible results from a binary classification algorithm are possible: true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN), and the confusion matrix captures the same. While, Accuracy (TP+TN)/(TP+TN+FP+FN), Sensitivity TP/(TP+FN), and Specificity TN/(TN+FP) are typical statistical indicators directly calculated from the confusion matrix. These are explained using Fig. 10. Typically, overall accuracy can identify how good the classifier is at classifying the TP and TN images, but it may not be able to identify individual performance. The sensitivity and specificity are able to capture the same.

C. Numerical Results

Based on the synthesized cases, we have divided this subsection into two parts:

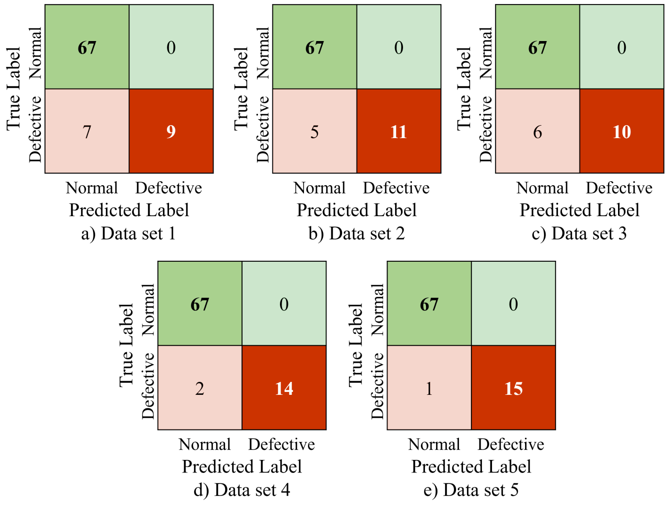
Case A: As a part of this case, we would like to observe how increasingly added GAN-generated synthetic images and consequent reduction in skewness impacts the overall performance of the classifier. Given that our test samples are true data sets, the impacts of synthetic data on test results are eliminated. The challenges with skewed data sets are imminent in Fig. 11, where, it can be seen that the classifier is able to classify normal insulators with 100% accuracy but struggles with defective insulators. Here, multiple defective insulators are classified as non-defective (False positive). We can observe that despite the "graininess" of the GAN-generated synthetic images and their use in training, the classification accuracy increases as the skewness continues to decrease. It can be seen in Fig. 12, increasing accuracy is not monotonically increasing, specifically for data set 3. However, by construction, we reused synthetic images across data sets, which implies that DCGAN can synthesize poor-quality images, which, when coupled with high skewness and low sample size for training, even with the addition of synthetic images, can result in a reduction in the classification accuracy. As shown in Table III, the discrepancy is visible across all the metrics.

Nevertheless, following the addition of DCGAN-synthesized images, the specificity increased from 0.56 to 0.94, and overall accuracy from 91.6% to 98.8%.

Case B: Although the datasets are quite balanced, the impacts of limited data available for training are visible in

TABLE III
DIFFERENCES IN MODEL PERFORMANCE BASED ON DATA SET

Data set	Sensitivity	Specificity	Overall Accuracy (%)
Data set 1	1.00	0.56	91.6
Data set 2	1.00	0.69	93.9
Data set 3	1.00	0.63	92.8
Data set 4	1.00	0.88	97.6
Data set 5	1.00	0.94	98.8



True \ Predicted	Normal	Defective
Normal	67	0
Defective	7	9

a) Data set 1

True \ Predicted	Normal	Defective
Normal	67	0
Defective	5	11

b) Data set 2

True \ Predicted	Normal	Defective
Normal	67	0
Defective	6	10

c) Data set 3

True \ Predicted	Normal	Defective
Normal	67	0
Defective	2	14

d) Data set 4

True \ Predicted	Normal	Defective
Normal	67	0
Defective	1	15

e) Data set 5

Fig. 11. Confusion matrix of the test data with models based on skewed data

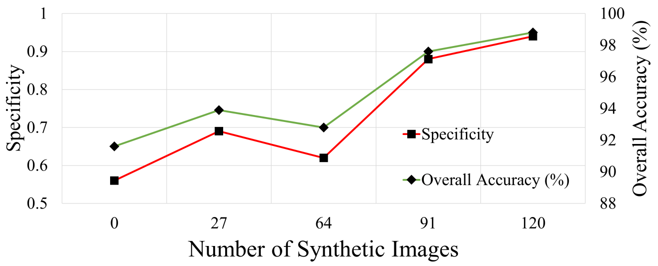


Fig. 12. Trends in specificity and overall accuracy

Fig. 13. Note that the metrics with DCGAN-generated and oversampling-based results are the same when we have used 36 normal and 36 defective images from training, and by construction, similar sets of data were used in both subcases. We also observe that with random oversampling, one may risk overfitting, and therefore, the metrics are lower in the case of random oversampling. We can also observe the sensitivity to be 100% across the board.

IV. CONCLUSION

This paper presents an end-to-end framework for improving the accuracy of line-post insulator fault classification through generative approaches to creating synthetic images. In particular, the focus would be the development of a ResNet-18 architecture-based machine vision algorithm utilizing UAV-captured images. However, insufficient insulator data availability and the underlying skewness within the available data make the development of the machine vision algorithm challenging, and in this regard, we sought the help of deep Convolutional Generative Adversarial Networks (DCGAN) to generate synthetic images. In this regard, we described the use of DCGAN in power distribution system automation and the training process for synthetic data generation. Following the training of the ResNet-18 architecture-based classifier, we observed how the introduction of the synthetic data increases the performance of the classifier. We also observed the performance improvement for the DCGAN-based image synthesizer compared to the random oversampling algorithm. We expect that this paper would help power engineers to understand the suitability of using generative approaches for power engineering applications. Given the crack formation, damages in the insulator are governed by underlying physical laws, it would be of interest to incorporate them while developing the GAN model as a part of future work.

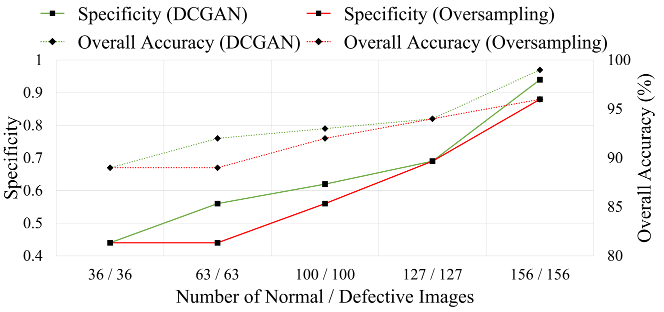


Fig. 13. Comparative metric-wise trends when data set are not skewed

REFERENCES

[1] V. S. Murthy, K. Tarakanath, D. K. Mohanta, and S. Gupta, "Insulator condition analysis for overhead distribution lines using combined wavelet support vector machine (svm)," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 17, no. 1, pp. 89–99, 2010.

[2] L. Matikainen, M. Lehtomäki, E. Ahokas, J. Hyyppä, M. Karjalainen, A. Jaakkola, A. Kukko, and T. Heinonen, "Remote sensing methods for power line corridor surveys," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 10–31, 2016.

[3] D.-Y. Lee, J.-O. Kim, J.-Y. Yun, and M.-H. Choi, "A development of an image diagnosis algorithm of distribution equipment using deep learning," *Proceedings of the KIEE Conference*, pp. 1504–1505, 2017.

[4] J. Katrasnik, F. Pernus, and B. Likar, "A survey of mobile robots for distribution power line inspection," *IEEE Transactions on Power Delivery*, vol. 25, no. 1, pp. 485–493, 2010.

[5] V. N. Nguyen, R. Jenssen, and D. Roverso, "Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning," *International Journal of Electrical Power Energy Systems*, vol. 99, pp. 107–120, 2018.

[6] J. Brownlee, *Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery, 2020.

[7] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11, 2003, pp. 1–8.

[8] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2016, p. 805–808.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016.

[11] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE transactions on knowledge and data engineering*, 2021.

[12] A. Dash, J. Ye, and G. Wang, "A review of generative adversarial networks (gans) and its applications in a wide variety of disciplines–from medical to remote sensing," *arXiv preprint arXiv:2110.01442*, 2021.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.