Data augmentation using generative adversarial networks for electrical insulator anomaly detection

by

Lei Luo

M.A., Kansas State University, 2017

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2020

Approved by:

Major Professor
William H. Hsu

Copyright

Abstract

Electricity has been an essential part of our life. Insulators, which are widely used for electricity transmission, are prone to be damaged and need constant maintenance. Traditionally, the inspection job is time-consuming and dangerous as workers would have to climb up the electricity tower.

Deep learning has offered a safe and quick way to inspections. About 3000 insulators images are taken from different angles using a drone. Due to great difference in number of good and damaged insulator, directly training a classifier on the imbalanced data lead to low recall value on the damaged insulators. Generative adversarial networks (GANs) were introduced as a novel way to augment data. However, traditional GANs are either incapable of generating high quality images or fail to generate minority class images when minority class examples are far less.

In this study, a novel GAN model, Balancing and Progressive GANs (BPGANs), was proposed for effectively making use of all classes information and generating high quality minority images at the same time. Results show that PGANs, StyleGANs, and BPGANs were able to generate high-resolution images and improve classification performance. PGANs achieved the better results than BPGANs. This may be because BPGANs only provides 2 additional latent codes since it is a binary classification, having little effect on generating desired images. BPGANs seemed to have difficulties generating class-specific images, which might be because that the classification loss is too little compared to the source loss and optimization was more focused to optimize the source loss. This indicates that learning representations of data progressively from low resolution to high resolution is an effective approach, however, embedding class label information in the fashion of AC-GANs and BGANs might not be appropriate for augmenting binary class data sets.

Table of Contents

List of Figures vi

Acknowledgements vii

1 Introduction 1

1.1 Overview 1

1.2 Motivation 2

1.3 Problem Statement 2

1.4 Objectives 3

2 Background 4

2.1 Deep Learning for Image Classification 4

2.1.1 Overview of Convolutional Neural Networks 5

2.1.2 AlexNet 5

2.1.3 VGG Nets 6

2.1.4 GoogLeNet 6

2.1.5 ResNet 7

2.1.6 DenseNet 8

2.2 Generative Adversarial Networks 10

2.2.1 GANs for Generating High Resolution Images 10

2.2.2 Conditional GANs 11

3 Methodology 13

3.1 Embedding Class Label 13

3.2 Generating High Quality Images . . . 14
3.3 BPGANs . . . 14

4 Experiments . . . 16
4.1 Data set . . . 16
4.2 DNNs for Classifying Insulators . . . 17
4.3 BGANs . . . 18
4.4 AC-GANs . . . 18
4.5 PGANs . . . 18
4.6 StyleGANs . . . 19
4.7 BPGANs . . . 19

5 Results . . . 20
5.1 Results on GANs . . . 20
5.1.1 BGANs . . . 20
5.1.2 AC-GANs . . . 21
5.1.3 StyleGANs . . . 21
5.1.4 PGANs . . . 21
5.1.5 BPGANs . . . 24
5.2 Image Qaulity Assessment . . . 24
5.3 Image Diversity Assessment . . . 25
5.4 Classification Results . . . 27

6 Conclusion and Future Work . . . 29
6.1 Summary and Conclusions . . . 29
6.2 Future Work . . . 31

Bibliography . . . 32

List of Figures

2.1 Architecture of AlexNet 6

2.2 Architecture of VGG-19 7

2.3 Architecture of Inception Layer in GoogLeNet 8

2.4 Architecture of Residual Block 9

2.5 Architecture of DenseNet 9

2.6 Architecture of GANs 11

3.1 Architecture of BPGANs 15

4.1 Examples of Insulators 17

5.1 Damaged Insulator Examples Imagined by BGANs 21

5.2 Damaged Insulator Examples Imagined by AC-GANs 22

5.3 Damaged Insulator Examples Imagined by StyleGANs 23

5.4 Damaged Insulator Examples Imagined by PGANs 23

5.5 Damaged Insulator Examples Imagined by BPGANs 24

5.6 Successful Identification Rate of Damaged Insulators Gneerated by Different GANs 25

5.7 SSIM among Images Produced by GANs 26

5.8 Successful Identification Rate of Damaged Insulators Gneerated by Different GANs 27

5.9 Successful Identification Rate of Damaged Insulators Gneerated by Different GANs 28

Acknowledgments

Thanks to Black and Veatch for providing original insulator images and the K-State Laboratory for Knowledge Discovery in Databases for providing annotated images and computing power to conduct the experiments in this study. Thanks to my advisor for tremendous help in advising my research.

Chapter 1

Introduction

This chapter introduces a visual anomaly detection task in the general application area of power system component inspection, and discusses the issue of class imbalance and the need for data augmentation for this task. I then present two approaches, data balancing and style transfer, and discuss the problem of combining these approaches in a single generative model.

1.1 Overview

There is a huge demand for electricity in the U.S. In 2018 U.S. net electricity generation increased by 4% in 2018, reaching a record high of 4,178 million megawatthours (MWh), surpassing once again the pre-recession peak of 4,157 million MWh set in 2007. Both the residential (about 1,500 MWh) and commercial sectors (about 1,400 MWh) reached all-time highs for retail sales of electricity in 2018[1]. Power cuts can cause major losses of productivity, safety, and resources, and consequent economic loss, to all sectors of society. According to the report from the U.S. Energy Information Administration in 2013[2], between 2003 and 2012, an estimated 679 widespread power outages occurred due to severe weather. Power outages close schools, shut down businesses and impede emergency services, costing the economy billions of dollars and disrupting the lives of millions of Americans. Therefore,

regular inspections are required to prevent faults that cause power outage. Visual inspections on transmission lines is the common way to maintain electricity supply. They are usually carried out by skilled worker foot-patrolling or using helicopter-assisted methods[3], which is costly and risky.

Many efforts have been made to help inspect electrical transmission lines more efficiently. Drones and robots that are capable of patrolling are made to increase inspection safety and lower costs[4–6]. After images and videos are collected from robots, trained workers spend huge amounts of time to analyze these images in order to recognize any faults. This obviously is time-consuming and prone to human bias. To reduce the human bias and better classify insulators, new approaches should be explored.

1.2 Motivation

Significant advances have been seen in the field of deep learning in recent years. Various types of deep neural networks have achieved great success in various computer vision tasks, such as image classification[7–9] and object detection[10–12]. One of the drawbacks of deep learning methods is that they generally require huge amount of data in order to classify insulators. When training examples are not sufficient, there are several approaches to augmenting data, including image oversampling, random rotation, shifts, and etc. However, traditional data augmentation techniques do not improve the results significantly[13]. In addition to limited data sets, class imbalance also creates many problems for neural network classifiers[14]. Thus, it is necessary to propose new strategies to enrich data sets and mitigate data imbalance issues.

1.3 Problem Statement

In recent years, generative modeling, as a more promising approach to data augmentation, has emerged as an effective methodology. In particular, a type of neural network called *generative adversarial networks (GANs)* was first introduced in 2014[15]. Since then, various

GAN extensions were proposed, such as CycleGANs, DCGANs, and Progressively-Growing GANs[16–18]. GANs have shown superior results compared to traditional data augmentation techniques as they are capable of imaging different alterations to images such that they have a better understanding of them[14]. However, these GANs might not work well when data sets suffer from excessively severe class imbalance, because they are trained on the minority class examples for generating new examples, and GANs are not likely to learn well from a minority class that has very few examples. In order to mediate the class imbalance issue , another branch of research has been focusing on developing the conditional version of GANs[19–21], aiming to include class label in GANs training. However, these conditional version of GANs generally works well with relatively low resolution data set, such as images whose size are 64×64 or 128×128.

1.4 Objectives

To overcome the limitations of existing GANs, we need to assess the specific technical strengths of various GAN-based models for data augmentation and propose new approaches based on the limitations. Thus, the objectives of this study include:

- Train a classifier using deep neural network on the original data set, which is not augmented and is imbalanced, to classify damaged insulators against good insulators. The classification performance results can serve as a baseline.
- Train various GANs models to augment and balance the data set. Then, train the same classifier on the augmented data set and compare the classification performance against the baseline from the previous step.
- Propose and develop a new model that is able to make use of both majority and minority class information when training so that it works well even when the number of minority class examples is scarce. The model is also capable to generate high quality examples for all categories.

Chapter 2

Background

2.1 Deep Learning for Image Classification

Image classification can be defined as the task of assigning images to predefined categories. Traditionally, human-crafted features were first extracted from images using feature descriptors, such as statistical information about pixels. These features then served as input to trainable algorithms. The most obvious disadvantage of this approach is that the success of the classifier depends highly on the design of handcrafted features, which is usually a formidable task[22;23].

Major advances have been seen in the field of deep learning in recent years. Various types of deep neural networks (DNN) have achieved great success in image classification[7–9]. Compared to traditional classification approaches, DNNs are capable of automatically extracting features from images, and thus greatly reduce or eliminate dependence on expert-crafted features.

The most significant advance, which has captured intense interest in DNNs, especially for image classification tasks, was achieved in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012[24] by Krizhevsky et al. (2012).[25] They used a DCNN to classify approximately 1.2 million images into 1000 classes, with record-breaking results. Since then, DNNs have dominated subsequent versions of the ILSVRC and, more specifically, its image

classification component[26,27].

2.1.1 Overview of Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are feedforward networks where inputs and intermediate results flow in one direction from an input image to class prediction for image classification tasks. Even though CNNs vary in the architectures, CNNs consist of several common building blocks, including convolution, activation, pooling layers. These building blocks form a basic module that is often stacked on top of each other to form a deep model. The convolutional layers can be used as feature extractors and they are able to learn feature representations from input neurons. Each neuron in the feature map has a receptive field, which connects to neighborhood neurons from the previous neurons through connected weights. Neurons convolve with weights and are usually activated with non-linear activation functions. All neurons within a feature map have weights that are constrained to be equal. The pooling layers are used to reduce dimensions of the feature maps and achieve spatial invariance to input distortions and translations[28]. Average pooling was used to be widely used to propagate the average of input values, but more recent studies[25–27,29] showed that max pooling layers yield better results as they propagate the maximum value within a receptive field to the next layer. Fully connected layers are usually positioned at the last a few layers in CNN. Features are propogated via stacks of convolutional layers and pooling layers, and the connected layers that follow these layers interpret these feature representations and perform the function of high-level reasoning[26,27].

2.1.2 AlexNet

The AlexNet[7] was proposed by Alex Krizhevesky and others in 2012. It achieved state-of-the-art recognition accuracy against all the traditional machine learning and computer vision approaches in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It was a big breakthrough and turning point where interest in deep learning increased rapidly.

AlexNet consists of five convolutional blocks (See Figure 2.1), each of which performs

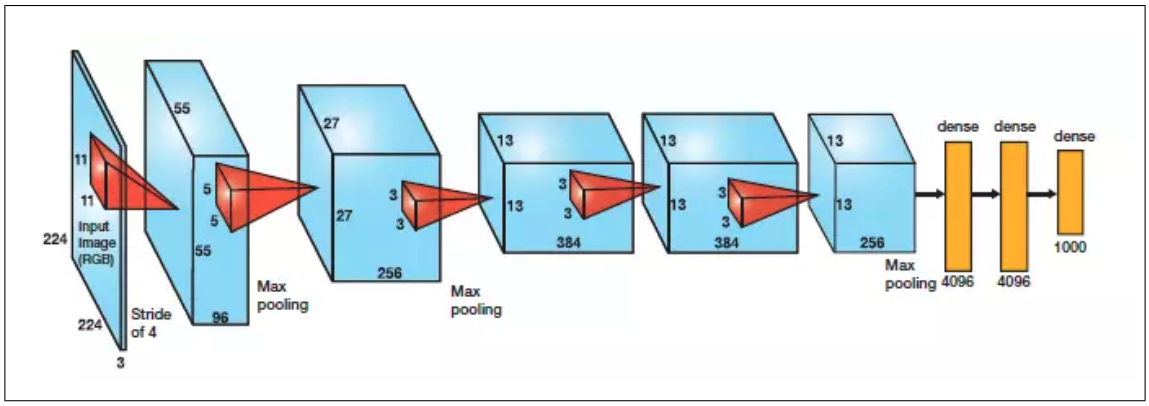

Figure 2.1: *Architecture of AlexNet*

convolution, max pooling, and local response normalization on the input. The Local Response Normalization (LRN) can be used in two ways. It can be applied on single channel or feature maps, where an $N \times N$ patch is selected from the same feature map and normalized based on the neighborhood values. LRN can also be applied across the channels or feature maps. Three convolutional blocks are followed by two fully connected layers. Similar to other CNN models that be applied to classify images, the last layer is a softmax layer.

2.1.3 VGG Nets

The Visual Geometry Group (VGG) achieved 2nd place in the 2014 ILSVRC[30]. The major contribution is that it shows the depth of CNN plays a vital role in order to achieve better recognition or classification accuracy. The VGG net features 2 convolutional layers with ReLU activation function, which is followed by a max pooling layer. The last four layers of VGG net are three fully connected layers and a single softmax layer classifying the images. There are three different sub-versions of VGG net, which are VGG-11, VGG-16, and VGG-19 (See Figure 2.2). They consist of 11, 16, 19 convolutional layers respectively.

2.1.4 GoogLeNet

GoogLeNet[8] won the 2014 ILSVRC[30]. The model was proposed with the purpose of reducing computation complexity compared to the traditional CNN. The model incorporates inception

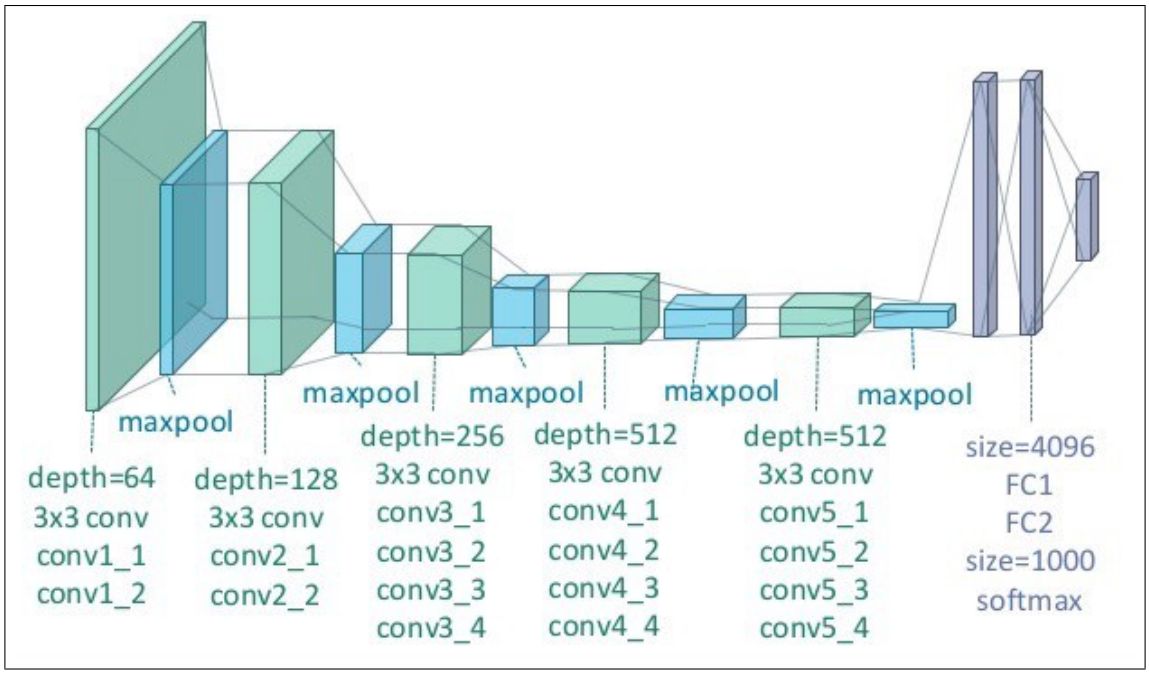

Figure 2.2: *Architecture of VGG-19*

layers which have variable receptive fields with different kernel sizes. These receptive fields is able to capture sparse correlation patterns in the new feature map stack. The idea of inception layer is that the previous layer goes through a 1×1 convolution, 3×3 convolution, 5×5 convolution and a 3×3 max pooling operations, each of the which is then concatenated to output the final result. Figure 2.3 shows the structure of inception layer. A stack of inception layers were used in GoogLeNet. The 1×1 convolution kernels allowed for dimension reduction before computationally expensive layers. GoogLeNet has 22 inception layers in total.

2.1.5 ResNet

The Residual Network (ResNet) architecture[9] won the first place in ILSVRC 2015. The idea of ResNet is to design very deep networks that do not suffer from the vanishing gradient problems. There are several different versions of ResNet, which have different number of layers. The major difference between ResNet and traditional CNN is the residual layer. The output of residual layer then goes through a set of common operations, including convolution with different sizes of filters and Batch Normalization (BN), followed by application of a

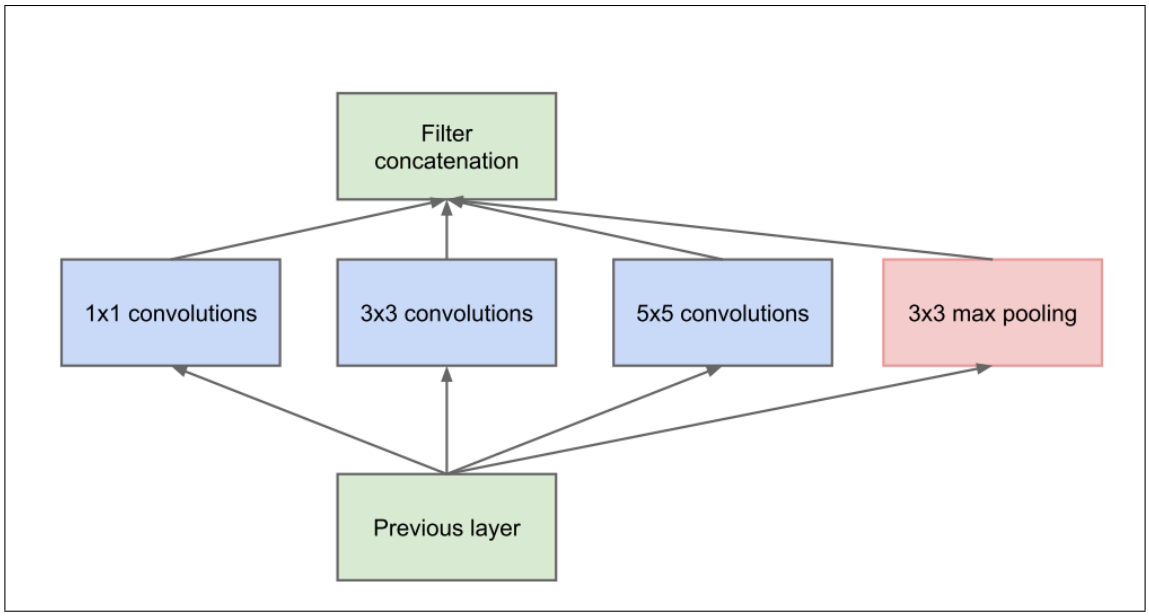

Figure 2.3: *Architecture of Inception Layer in GoogLeNet*

ReLU activation function. The final output of the residual layer is additively computed from the previous layer and the result from a series of operations. The entire ResNet consists of several residual blocks. Different modifications on the original ResNet were proposed. Zagoruvko et al.[31] proposed a wider version of ResNet. Aggregated residual transformation[32] was proposed to improve the ResNet architecture.

2.1.6 DenseNet

DenseNet[33] was proposed by Gao et al. The model has densely connected CNN layers, the output of which are concatenated with all successor layers in a dense block. This architecture is efficient for feature reuse, which can greatly reduce network parameters. It features several dense blocks and transition blocks, which are placed between two adjacent dense blocks. Figure 2.5 shows the architecture of DenseNet.

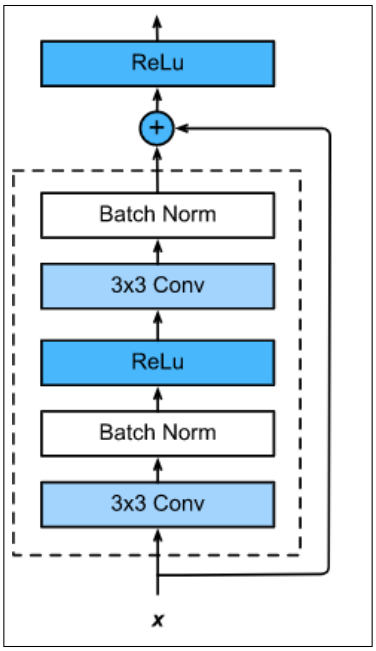

Figure 2.4: *Architecture of Residual Block*

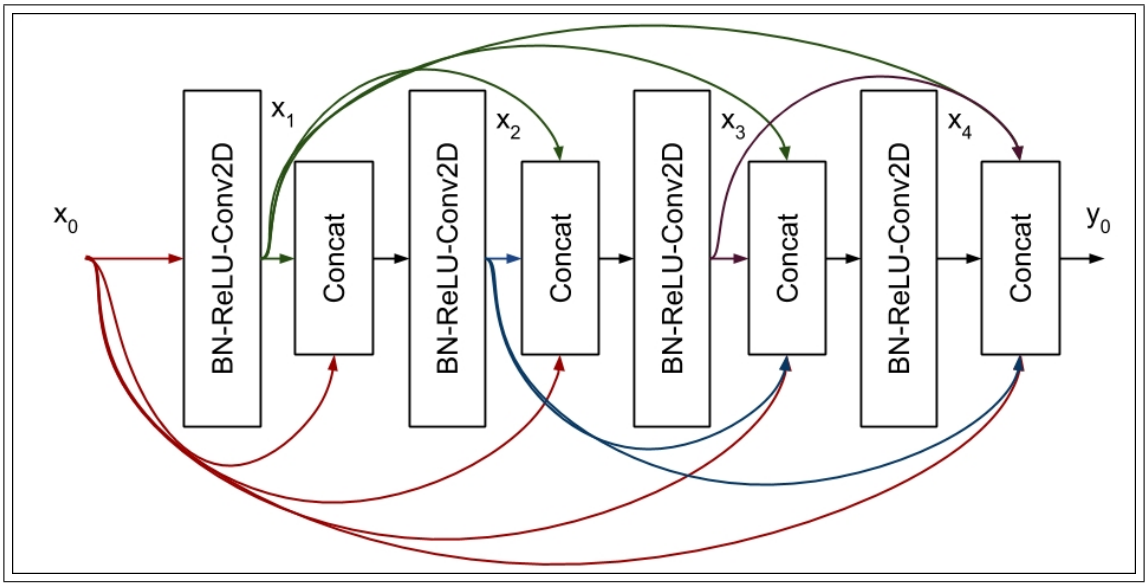

Figure 2.5: *Architecture of DenseNet*

2.2 Generative Adversarial Networks

Generative models are designed to model and reproduce the statistical distribution of the training data, allowing the synthes of data from the learned distribution. The key incentive behind GANs is estimating the underlying probability density or probability mass function of the observed data. GANs learn the probability distribution implicitly by computing the similarity of the distribution between the real training examples and the "fake" data generated by the learned model. After the model is trained well, it naturally can be used to generate data that have similar distribution as the real data.

GANs typically consist of a generator and a discriminator. The two compete against one another: the generator tries to fool the discriminator by producing "fake" data and the discriminator aims to distinguish the "fake" data from the real data. The general architecture is shown in Figure 2.1. The learning process is guided by a minmax game (See Equation 2.1) where the discriminator (D) desires to increase the probability of classifying images as real when they (x) are sampled from the real distribution ($p_{data}(x)$). Meanwhile, the generator (G) wishes to decrease the likelihood of generated images as real when images are sample from the "fake" distribution ($p_z(z)$). As learning progresses, the discriminator gets better at classifying the data being real or not, and the generator becomes better at producing "realistic" data. Naturally, the generator can then be used to generate data when training examples are not sufficient.

$$\min_G \max_D V(D,G) = \mathop{\mathbb{E}}_{x \in p_{data}(x)}[logD(x)] + \mathop{\mathbb{E}}_{z \in p_z(z)}[log(1 - D(G(z)))] \tag{2.1}$$

2.2.1 GANs for Generating High Resolution Images

GANs have difficulties in generating high resolution images as they make it easier for the discriminator to distinguish the "fake" images among the training data. High-resolution data also prevents from using larger minibatches due to GPU memory limitations and thus com-

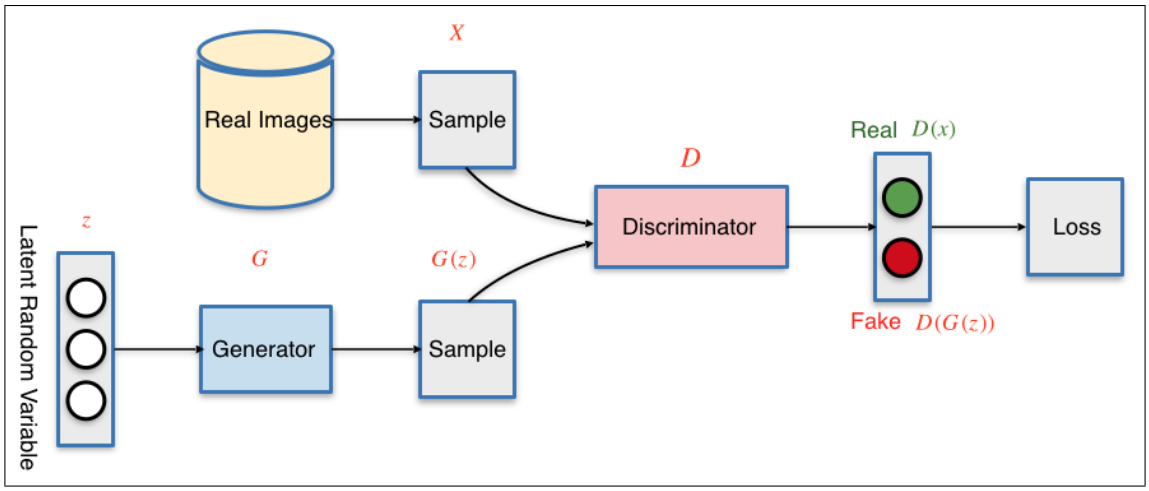

Figure 2.6: *Architecture of GANs*

promising training stability[18;19]. Based on the work of Wang et al. (2017)[34], Durugkar et al. (2016)[35], and Hierarchical GANs[36–38], Karras et al. (2018)[18] proposed progressively-growing GAN, whose key idea is to grow both the generator and discriminator progressively, starting from easier low-resolution images, and add new layers that introduce higher-resolution details. This speeds up training and improves stability in high resolution images. Following the trace of progressively-grow GAN and the idea of style transfer, Karras et al. (2019)[39] proposed a style-based generator for GANs. It features an automatically learned, unsupervised separation of high-level attributes and stochastic variation in the generated images, which enables intuitive, scale-specific control of the image synthesis.

2.2.2 Conditional GANs

Another branch of research focuses on the conditional version of GANs[20;21;40], which aims to embed label information into GANs training process. Conditional GANs[20] incorporate labels into both generator and discriminator by modelling the conditional probability of $logD(x|y)$ and $log(1-D(G(z|y)))$. As a result, the objective function in Equation 1 is slightly modified into the conditional form (See Equation 2.2).

$$\min_{G} \max_{D} V(D,G) = \mathop{\mathbb{E}}_{x \in p_{data}(x)} [logD(x|y)] + \mathop{\mathbb{E}}_{z \in p_z(z)} [log(1 - D(G(z|y)))] \tag{2.2}$$

Mariani et al,.(2018) proposed Balancing GANs (BGANs)[21] as an augmentation tool to restore balance in imbalanced data sets. They argued that the few minority-class examples may not be enough to train a GAN, so they incorporated all available images of majority and minority classes. BGANs try to achieve class balance by applying class conditioning in the latent space to drive the generation process towards the target class. BAGAN features an autoencoder that learns an accurate class-conditioning in the latent space and then initializes the generator with the encoder of the autoencoder.

Odena et al. 2017 proposed Auxiliary Classifier GANs(AC-GANs)[41], which models the probability distribution of class labels in addition to the original objective of GANs. The objective of AC-GANs consists of two parts: the loglikelihood of the correct source, L_S, and the log-likelihood of the correct class, L_C:

$$L_S = E[logP(S = real|X_{real})] + E[logP(S = fake|X_{fake})] \tag{2.3}$$

$$L_C = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})] \tag{2.4}$$

D is trained to maximize $L_S + L_C$ while G is trained to maximize $L_C - L_S$. AC-GANs learn a representation for z that is independent of class label.

Chapter 3

Methodology

As mentioned in Chapter 1, existing GANs cannot produce high resolution images and embed class label information at the same time. In order to address this limitation, I proposed a new GAN model, which I refer to as Balancing and Progressively Growing GANs (BPGANs) after the existing indpendent classes of Balancing GANs (BGANs) and Progressive Growing of GANs (PGANs).

3.1 Embedding Class Label

Through Conditional GANs[20], a novel method of embedding class labels was developed. Conditional GANs incorporate labels into both the generator and the discriminator by modelling the conditional probability of $logD(x|y)$ and $log(1 - D(G(z|y)))$ (See Equation 2.2). BPGANs follow the same method on incorporating class label information. In the work of AC-GANs[41], the model embeds the class label information by having an extra classifier to predict which class images that are fed to discriminator belongs to. Inspired by this AC-GANs, BPGANs will adopt similar approaches.

3.2 Generating High Quality Images

Karras et al. (2018)[18] proposed Progressive Growing of GANs (PGANs), whose key idea is to grow both the generator and discriminator progressively, starting from easier low-resolution images, and add new layers that introduce higher-resolution details. This speeds up training and improves stability in high resolution images. Following the trace of PGANs, the BPGANs model adopts the same strategy.

3.3 BPGANs

Motivated by AC-GANs and PGANs, the proposed BPGANs model will model class conditional distribution as well as generating more resolution images. The architecture of BPGANs is shown in Figure 3.1, where both generator and discriminator both are trained from low resolution to a resolution of 512×512 and label information is provided during training.

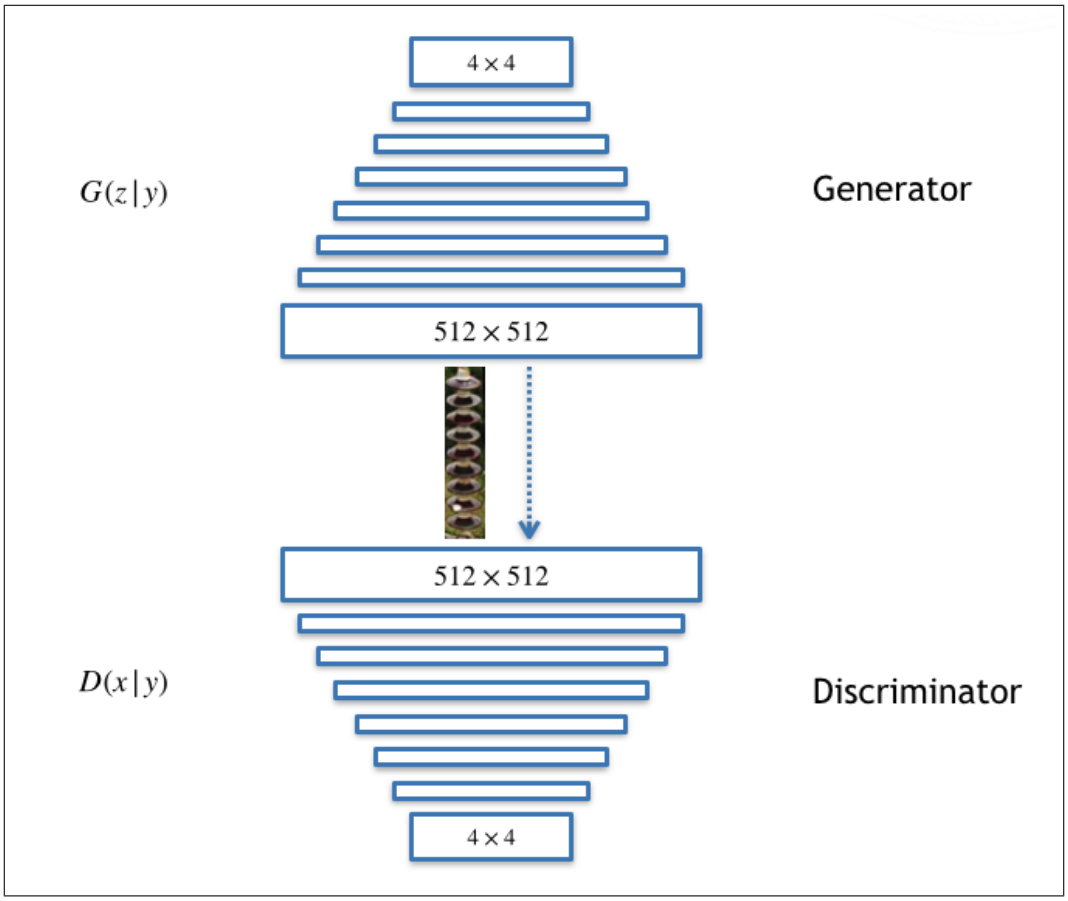

Figure 3.1: *Architecture of BPGANs*

Chapter 4

Experiments

In this chapter, I discuss the data set, how to split data set into training, validation, and test data set. A deep neural network was trained on the non-augmented data setas a classifier to distinguish good insulators against damaged insulators. This classifier will be used as the baseline for comparing the performance of various GANs on data augmenting. Next, several GANs will be trained to augment existing insulators data set. In the end, I train the proposed BPGANs model. In the next chapter, I will compare resutls of different GANs model.

As mentioned in Chapter 1, existing GANs cannot produce high resolution images and embed class label information at the same time. In order to address this limitation, a new GANs model, which I call Balancing and Progressively Growing GANs (BPGANs), is proposed.

4.1 Data set

Drones, specifically unmanned aerial vehicles (UAVs), were used to take images of insulators. Some examples of insulators are shown in Figure 4.1. Raw insulator images were then annotated with each individual insulator being labeled as damaged or undamaged. Next, images were cropped and 3,861 individual insulators were obtained in total, among which

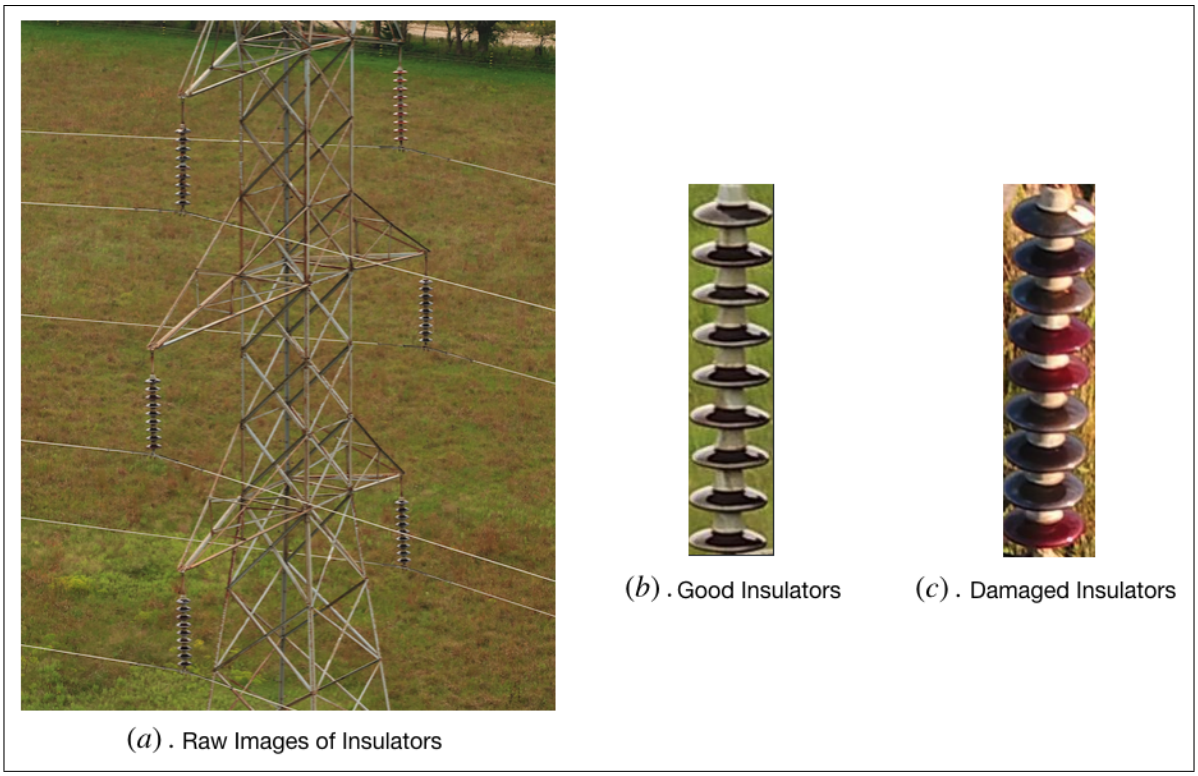

Figure 4.1: *Examples of Insulators*

2,972 are good insulators and the rest of 989 insulators are damaged. 80% of images were used as training data set and the rest was equally split as validation and test data set.

4.2 DNNs for Classifying Insulators

To establish a baseline, I trained a simple DNN on the original non-augmented data set. The DNN model consists of 9 convolutional blocks, before each one of which batch normalization was applied on the input. THe ReLU activation function was used after each convolution layer, and softmax was applied to reduce the dimensions of the output. The model was trained using stochastic gradient descent (SGD) on with batches of size 16, with the momentum set to 0.9 and the learning rate to 0.001.

4.3 BGANs

As discussed in the previous chapter, a VAE was first trained to obtain a dense representation, which was then used for sampling for the latent vector when training the BGANs model. Each VAE was trained for 4,000 iterations and Adam optimizer was used with learning rate of 0.001.

After the VAEs were trained, latent vectors of size 512 are sampled from a normal distribution with means and standard deviations of damaged and good insulators. The BGANs model was trained for 10,000 iterations and Adam optimizer was used to optimize the parameters with learning rate set to 0.002.

4.4 AC-GANs

AC-GANs not only supplies both the generator and discriminator with class labels, but also includes a classifier to classify the image category. This model produces good results and appears to stabilize training compared to the standard GAN formulation. Adam optimizer was used to train the model with initial learning rate set to 0.0001 and decayed after 200 iterations. The batch size, largely determined by the available GPU memory, is set to 16. The model was trained for 2,000 epochs.

4.5 PGANs

The key idea of PGANs is to grow both the generator and discriminator progressively, starting from easier low-resolution images (4×4), and add new layers that introduce higher-resolution details (eventually 1024×10244). This speeds up training and improves stability in high resolution images. The model was trained on batches of 4 images. The final resolution of images is 512×512, so there is one less scale training compared to the orginal PGANs implementation whose final resolution is 1024×10244. The model was trained at resolution from 4×4, 8×8, up to 512×512, and each was scale was trained with 48,000, 96,000, 96,000,

..., 96,000 iterations respectively.

4.6 StyleGANs

StyleGANs is proposed based on PGANs. The modification is that instead of feeding the original latent vector into the generator, a block of dense layers are applied on the latent vector to learn what the authors called the intermediate latent vectors, and then the intermediate latent vectors are fed into the generator. At each scale of training, an affine transformation was applied on the intermediate latent vector as a way to learn some sort of style before feeding into the generator. The architecture of the generator and discriminator of StyleGANs are same of PGANs. The model was trained on batches of 7 images. Adam optimizer was used with learning rate set to 0.0015 and decayed after 200 iterations.

4.7 BPGANs

BPGANs combine the advantages from AC-GANs and PGANs, which is able to embed class label information into training as well as producing high resolution images. Similar to AC-GANs, BPGANs embed class labels by including an extra classifier. The generator and discriminator progressively grow from an image size of 4×4 up to 512×512. The training settings for the BPGANs are the same as those for the PGANs.

Chapter 5

Results

In this section, I discuss the results of different models from the previous chapter, including a classifier trained on the original, unbalanced data set and results from several GANs models. In order to evaluate the images generated by GANs qualitatively and quantitatively, similarity indices were calculated to evaluate how diverse and clear of the generated images are.

GANs were trained with the purpose of augmenting the unbalanced data set, so different GANs were then used to balance the original data set. Next, the same classifier was trained on different versions of augmented data set and classification results were compared against the original data set.

5.1 Results on GANs

5.1.1 BGANs

The final output image resolution is 512×512, and the generated images from BGANs are very blurry as BGANs are originally designed to produce image size of 64×64. Several examples of generated damaged insulators are shown in Figure 5.1.

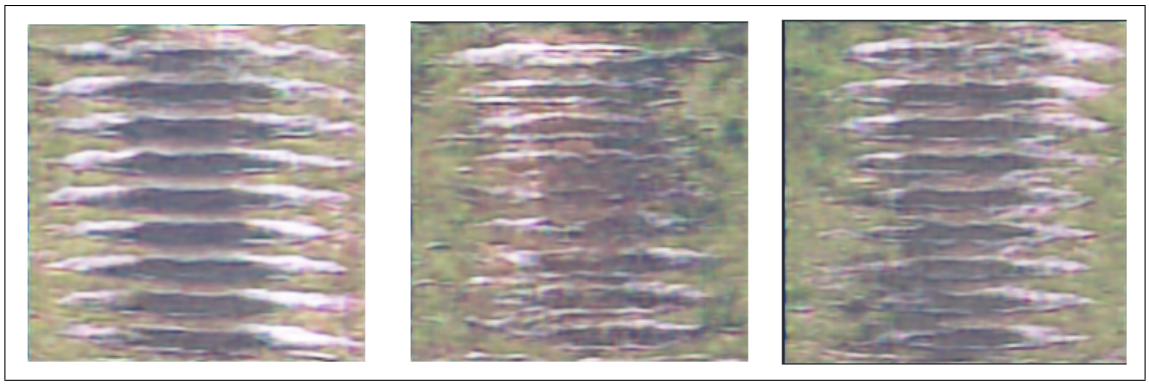

Figure 5.1: *Damaged Insulator Examples Imagined by BGANs*

5.1.2 AC-GANs

Three examples of damaged insulators produced by AC-GANs are shown in Figure 5.2. AC-GANs produced images that have better quality than that of BGANs, but they are still blurry and difficult. to distinguish damaged versus good insulators. This might be owing to that the classification loss was dominated by the source loss and much optimization was done to decrease the source loss.

5.1.3 StyleGANs

Compared to images generated by BGANs, StyleGANs was able to produce images with much higher quality. Four examples are shown in Figure 5.3, where the first three images are of better quality and the last one is one of the poorly-produced image examples.

5.1.4 PGANs

PGANs was able to produce images with similar quality as StyleGANs. Four examples are shown in Figure 5.4, where the first three images are of better quality and the last one is one of the poorly-produced image examples. Compared to images generated by StyleGANs, PGANs seem to be able to produce images with better image quality, where flash-overs are more clear to spot.

Figure 5.2: *Damaged Insulator Examples Imagined by AC-GANs*

Figure 5.3: *Damaged Insulator Examples Imagined by StyleGANs*

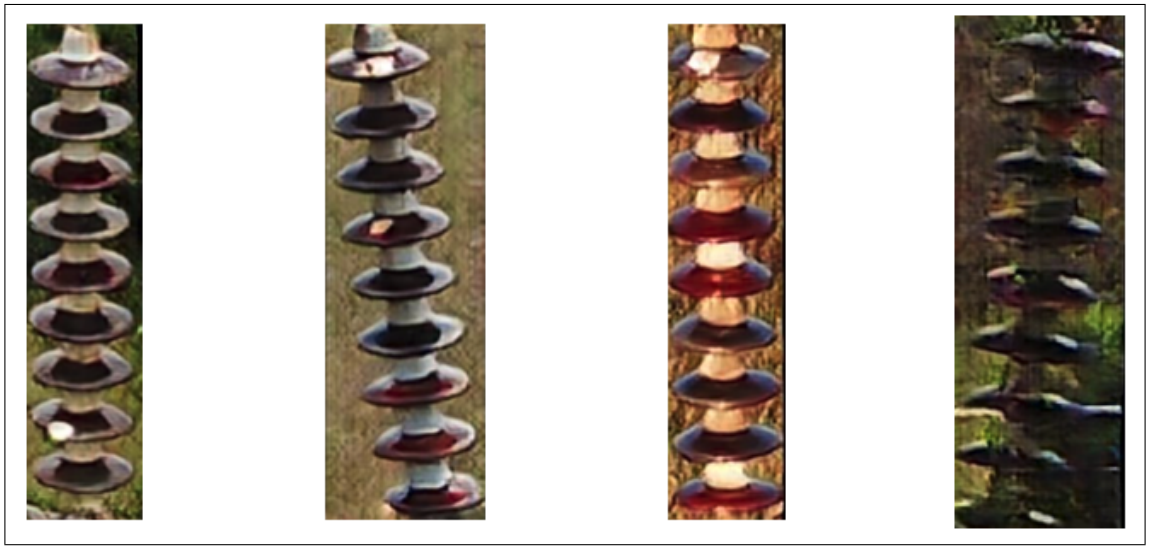

Figure 5.4: *Damaged Insulator Examples Imagined by PGANs*

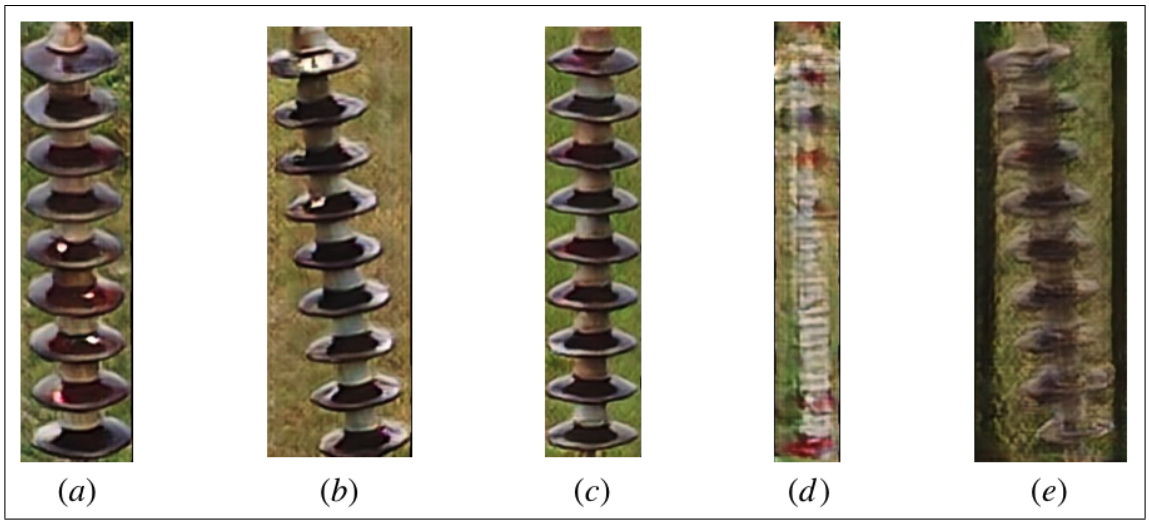

Figure 5.5: *Damaged Insulator Examples Imagined by BPGANs*

5.1.5 BPGANs

BPGANs produced mixed results. Some damaged insulators are shown in Figure 5.5. (a) and (b) look like damaged insulators. (c) seems to be a good insulators and (d) and (e) are pretty blurry. The reason that quite a few damaged insulators imagined by BPGANs look like good insulators might be because the classification loss is too small in magnitude compared to source loss, which measures the difference between real or fake images shown to the discriminator. Compared to StyleGANs and PGANs, BPGANs also tend to produce more blurry images.

5.2 Image Qaulity Assessment

In order to assess the quality of generated images from GANs, the images were evaluated by the trained classifier. The rational behind this assessment strategy is that what percentages of GANs-generated damaged data the classifier thinks are actually damaged. The results from different GANs are summarized in Figure 5.6. From the graph we can see that the trained classifier only classify nearly 30% of damaged insulators produced by BGANs as damaged, which makes sense as the images quality of the generated images are very blurry.

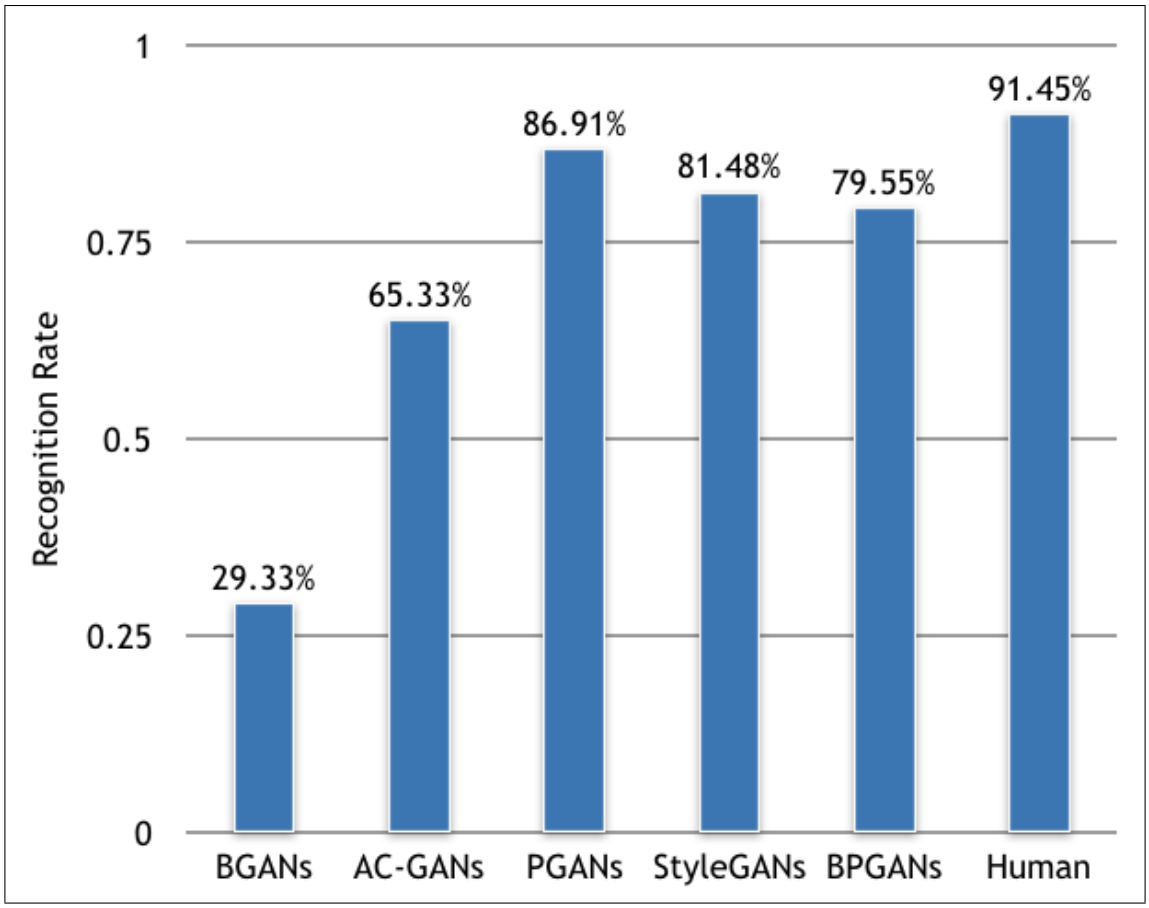

Figure 5.6: *Successful Identification Rate of Damaged Insulators Gneerated by Different GANs*

StyleGANs and PGANs produce much better image quality, and as a result more images are recognized as damaged by the classifier. The results showed that PGANs produced the best image quality among all four GANs models.

5.3 Image Diversity Assessment

In order to assess the images generated by GANs visually, I verified whether the generated images are merely copies of the training images and if the images have few modes, meaning they are close to one another. Two comparison schemes were adopted. First, I compared how similar the generated images are to the training set by randomly selecting 100 produced images, and then selecting the 100 closest images to the training set by calculating the

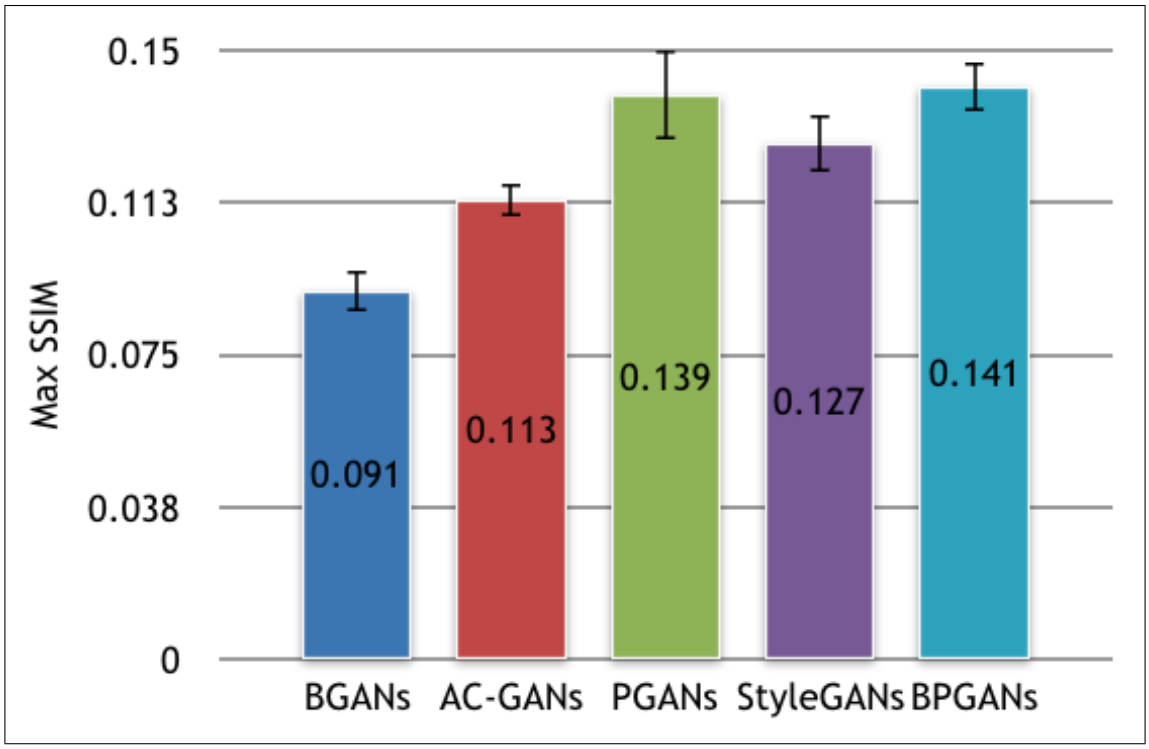

Figure 5.7: *SSIM among Images Produced by GANs*

structural similarity (SSIM) index. Finally, I calculated the mean value of SSIM values.

Next, 100 images were selected randomly from the generated poll. 100 closet images to the training set were chosen by calculating the SSIM values. Then, 100 closest images of the selected real images to the training set were selected. Eventually, the two sets of SSIM values were compared.

Figure 5.7 showed that maximum value and standard error of SSIM for different GANs. The results showed that the similarities among different sets of generated images are very low, indicating the modes of the images are of great diversity. Figure 5.8 showed that the images produced by GANs are not close to the training images, and thus not simply copies of the training set.

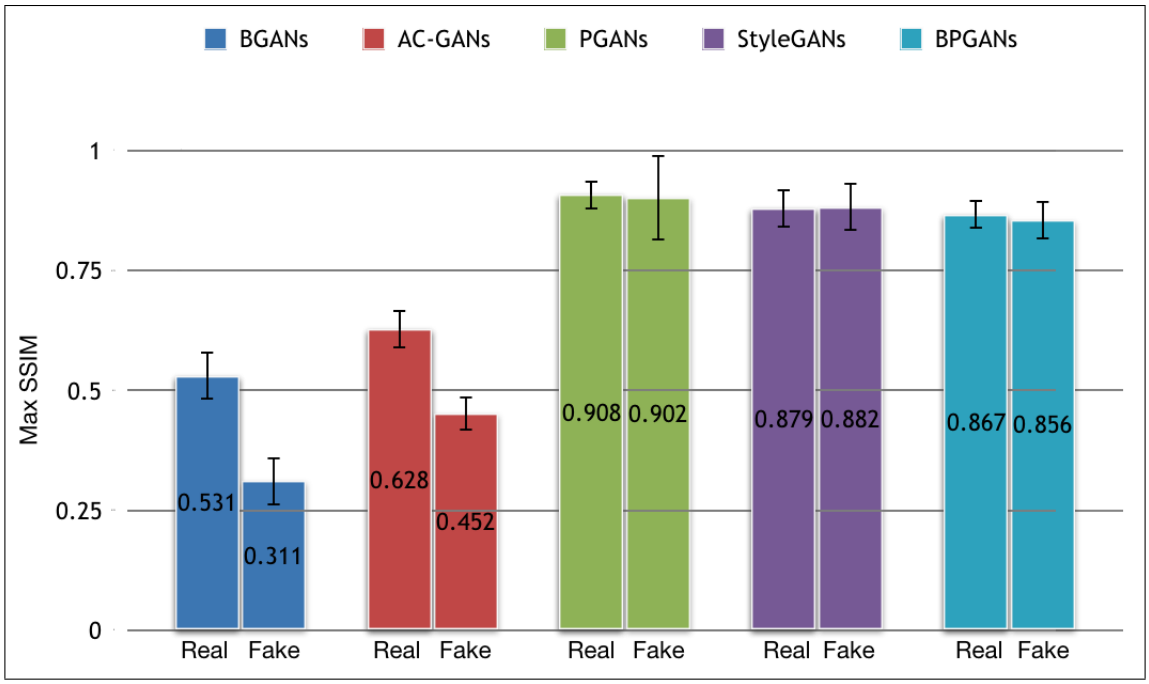

Figure 5.8: *Successful Identification Rate of Damaged Insulators Gneerated by Different GANs*

5.4 Classification Results

The ultimate goal of this study was to utilize GANs to augment and balance the data set in hope that a better classifier can be obtained to distinguish damaged insulators and good insulators. Therefore, the final assessment would be to train the same classifier on the augment the data set, observing if the classification performance would increase or not.

Figure 5.9 shows that the f1 scores and classification accuracy improved for all GANs, expect for the BGANs, compared to the classifier trained on the original classifier. Results show that classifier trained on data set augmented by PGANs achieved highest f1 score and accuracy.

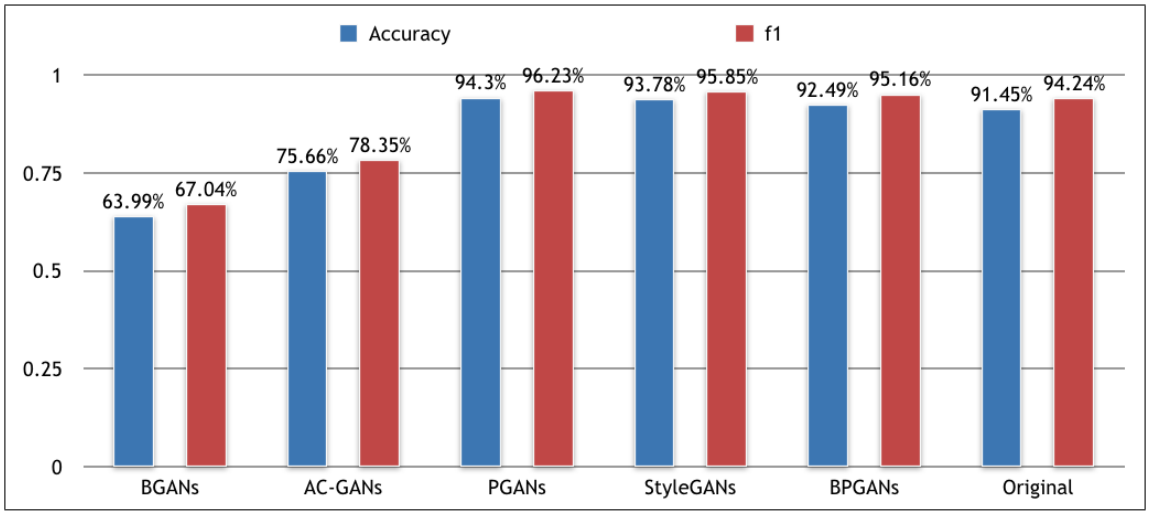

Figure 5.9: *Successful Identification Rate of Damaged Insulators Gneerated by Different GANs*

Chapter 6

Conclusion and Future Work

6.1 Summary and Conclusions

This work has introduced Balancing and Progressively Growing GANs (BPGANs), a new hybrid architecture for generative adversarial modeling of visual anomalies, that combines the problem of class balancing from that of transferring surface detail by class. Results applying BPGANs to the domain of electrical insulator inspection, where some types of damage can be determined from outdoor external views, suggest potential for BGAN/PGAN hybridization in visual anomaly detection.

Power systems are of great importance to mechanization across all aspects of society, and thus regular electrical inspections are needed. Recent advances in deep learning has offered viable approaches to automate inspection jobs. However, data availability can restrict how deep learning can be successful applied. Generative adversarial networks are capable of learning good representations of images and generate newly imaged ones based on the learned representation. Therefore, in this study, extensive experiments have been conducted on generating images, especially damage insulators, based on GANs.

Results show that BGANs and AC-GANs are able to combine class label during training, and thus provide more information guiding the generation of damaged insulators. However, they are designed to produce relative low-resolution images, and thus images generated are

blurry and even sabotage the classifier when augmenting the data set with images produced by them. StyleGANs and PGANs are capable to generate images of much higher quality because of the hierarchical architecture in both generator and discriminator. However, they are both trained only on the damaged insulators, and therefore does not make any use of information from the good insulators. Motivated by the advantages and drawbacks of existing GANs models, BPGANs is proposed. BPGANs is able to make full use of label information to generate images with good quality.

StyleGANs, PGANs, and BPGANs are able to produce damaged insulators examples with high quality. PGANs produced the best quality images with recognition rate about 87%, followed by StyleGANs (81%) and BPGANs (80%). The rest of GANs models failed to produce good quality images, having low recognition rate less than 70%. Low SSIM values by all the GANs models indicated that generated images are of great mode and are not similar to training data set. The classification accuracy and F1-score trained on the unaugmented data set is 91.45% and 0.9424. After augmenting the original data set, the classifier achieved higher F1-scores and classification accuracy. Highest improvement was seen on results from PGANs, with classification accuracy increased by about 3% and F1-score by 2%. StyleGANs and BPGANs also improved classification performance. The former achieved accuracy of 93.78% and F1-score of 0.9585, and the latter had accuracy of 92.49%, F1-score of 0.9516. BGANs and AC-GANs degraded the classification performance compared to the baseline as images generated by the two were of low quality.

The reason why BPGANs had difficulties distinguishing good versus damaged insulators might be because the classification loss is too little in magnitude compared to the source loss, which tried to tell if the images shown to the discriminator are from real or fake images. As a result, the optimizer focused more on optimizing the parameters to reduce the source loss. Another reason might be because there are only two class labels, which only provide 2 additional latent codes compared to original 512 latent codes, having limited effect on the model. Results indicated that learning representations of data progressively from low resolution to high resolution is an effective approach, however, embedding class label information in the fashion of AC-GANs and BGANs might not be appropriate for

augmenting binary class data sets.

6.2 Future Work

Embedding class label in forms of latent codes might not be suitable for augmenting data set with binary class label. Furthermore, in the training of GANs, latent vector is sampled from some distribution (for example, normal distribution) and then is used to produce fake images to fool the discriminator. Another way to include class labels is via the latent vector, on which future work is indicated, as this affects the convergence rate of the training as well as the image quality.

Bibliography

[1] Maggie Woodward. *Record U.S. Electricity Generation in 2018 Driven by Record Residential, Commercial Sales*. U.S. Energy Information Administration, 2019.

[2] Executive Office of the President. *Economic Benefits of Increasing Electric Grid Resilience to Weather Outages*. U.S. Department of Energy, 2013.

[3] Van Nhan Nguyen, Robert Jenssen, and Davide 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*, 99:107 – 120, 2018. ISSN 0142-0615. doi: https://doi.org/10.1016/j.ijepes.2017.12.016. URL http://www.sciencedirect.com/science/article/pii/S0142061517324444.

[4] J. Katrasnik, F. Pernus, and B. Likar. A survey of mobile robots for distribution power line inspection. *IEEE Transactions on Power Delivery*, 25(1):485–493, Jan 2010. ISSN 0885-8977. doi: 10.1109/TPWRD.2009.2035427.

[5] M. A. Gulzar, K. Kumar, M. A. Javed, and M. Sharif. High-voltage transmission line inspection robot. In *2018 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–7, Feb 2018. doi: 10.1109/ICEET1.2018.8338632.

[6] C. M. Shruthi, A. P. Sudheer, and M. L. Joy. Dual arm electrical transmission line robot: motion through straight and jumper cable. *Automatika*, 60(2):207–226, 2019.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/

4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[10] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. *CoRR*, abs/1901.07518, 2019. URL http://arxiv.org/abs/1901.07518.

[11] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. URL http://arxiv.org/abs/1506.01497.

[12] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. URL http://arxiv.org/abs/1506.02640.

[13] Marko Arsenovic, Mirjana Karanovic, Srdjan Sladojevic, Andras Anderla, and Darko Stefanovic. Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry*, 11(7), 2019. ISSN 2073-8994. URL https://www.mdpi.com/2073-8994/11/7/939.

[14] Joffrey L. Leevy, Taghi M. Khoshgoftaar, Richard A. Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42,

Nov 2018. ISSN 2196-1115. doi: 10.1186/s40537-018-0151-6. URL https://doi.org/10.1186/s40537-018-0151-6.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[16] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL http://arxiv.org/abs/1703.10593.

[17] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06434.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL http://arxiv.org/abs/1710.10196.

[19] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2642–2651. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305890.3305954.

[20] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

[21] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and A. Cris-

tiano I. Malossi. BAGAN: data augmentation with balancing GAN. *CoRR*, abs/1803.09655, 2018. URL http://arxiv.org/abs/1803.09655.

[22] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29:1–98, 06 2017. doi: 10.1162/NECO_a_00990.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. doi: 10.1109/5.726791.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL http://arxiv.org/abs/1409.0575.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[26] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

[28] Marc' Aurelio Ranzato, Y-Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. In *Proceedings of the 20th International Conference on Neural*

Information Processing Systems, NIPS'07, pages 1185–1192, USA, 2007. Curran Associates Inc. ISBN 978-1-60560-352-0. URL http://dl.acm.org/citation.cfm?id=2981562.2981711.

[29] Dan C. Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. High-performance neural networks for visual object classification. *CoRR*, abs/1102.0183, 2011. URL http://arxiv.org/abs/1102.0183.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.

[32] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. URL http://arxiv.org/abs/1611.05431.

[33] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

[34] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. LR-GAN: layered recursive generative adversarial networks for image generation. *CoRR*, abs/1703.01560, 2017. URL http://arxiv.org/abs/1703.01560.

[35] Ishan P. Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *CoRR*, abs/1611.01673, 2016. URL http://arxiv.org/abs/1611.01673.

[36] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015. URL http://arxiv.org/abs/1506.05751.

[37] Xun Huang, Yixuan Li, Omid Poursaeed, John E. Hopcroft, and Serge J. Belongie.

Stacked generative adversarial networks. *CoRR*, abs/1612.04357, 2016. URL http://arxiv.org/abs/1612.04357.

[38] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1612.03242, 2016. URL http://arxiv.org/abs/1612.03242.

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. URL http://arxiv.org/abs/1812.04948.

[40] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. URL http://arxiv.org/abs/1606.03657.

[41] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. *arXiv e-prints*, art. arXiv:1610.09585, Oct 2016.