

From Oversampling to DCGAN: A Systematic Comparison of Data Synthesis for Imbalanced Supervised Classification in Additive Manufacturing Processes

Jaehyun Kim¹, Hyungsoo Kim¹, Woojin Lee¹, Yubin Jung¹

¹Department of Industrial Engineering, College of Engineering, Hanyang University

The code is available at [GitHub](#)

Abstract

적층 제조(Additive Manufacturing, AM) 공정에서의 결함 분류는 비정상 상태의 희소성과 높은 데이터 수집 비용으로 인해 심각한 클래스 불균형 문제를 갖는다. 본 연구는 합성된 소수 클래스 이미지의 품질 차이가 지도학습 기반 결함 분류 성능에 미치는 영향을 체계적으로 분석하는 것을 목표로 한다. 정상 클래스와 두 가지 결함 클래스(Underfill_FR, Underfill_Fan)로 구성된 3D 프린팅 표면 이미지 데이터셋을 사용하여, 소수–다수 클래스 비율이 0.1 이하인 극단적 불균형 환경을 구성하고 전통적인 oversampling 기법(SMOTE, Borderline-SMOTE, ADASYN)과 GAN 계열 생성 모델(GAN, DCGAN)을 동일한 CNN 분류기와 평가 지표 하에서 비교하였다. 실험 결과, 전통적인 oversampling 기법은 제한적인 성능 향상에 그친 반면, DCGAN은 F1-score 를 0.7369에서 0.8611로, recall 을 0.7467에서 0.8667로 유의미하게 개선하였다. 반대로 완전연결층 기반 GAN은 full-shot 학습 환경에서도 의미 있는 고해상도 AM 이미지를 생성하지 못해 구조적 한계를 보였다. 정성적 분석을 통해 interpolation 기반 oversampling 은 결함 판별적 특징을 회색시키는 반면, DCGAN 은 mode collapse 에도 불구하고 결함 특유의 명암 및 표면 텍스처를 보다 효과적으로 보존함을 확인하였다. 또한 MNIST 보조 실험을 통해 극단적 데이터 희소성 환경에서 GAN 기반 합성의 민감성과 근본적 한계를 추가 검증하였다. 본 연구는 적층 제조 결함 학습을 위한 실용적 가이드라인을 제시하며, 극심한 데이터 부족 상황에서는 전통적 oversampling 이 기준선으로 유효하고, 최소한의 결함 샘플이 확보된 경우에는 합성곱 기반 DCGAN 이 보다 효과적인 대안임을 보인다.

1. Introduction

현대 기계 학습 (Machine Learning)과 딥러닝 (Deep Learning) 기술은 다양한 산업 분야에서 혁신적인 성과를 거두고 있으나, 데이터의 본질적인 분포 문제인 ‘클래스 불균형 (Class Imbalance)’은 여전히 해결해야 할 난제로 남아 있다. 실제 현실 세계의 애플리케이션, 예를 들어 제조 공정의 결함 탐지 (Anomaly Detection in Manufacturing), 의료 진단 (Medical Diagnosis), 금융 사기 탐지 (Fraud Detection), 그리고 네트워크 침입 탐지 (Intrusion Detection) 등에서는 정상데이터 (Majority)가 비정상 또는 결함데이

터 (Minority)에 비해 압도적으로 많은 불균형 현상이 빈번하게 발생한다.

일반적인 지도 학습 (Supervised Learning) 알고리즘은 전체 데이터셋의 정확도를 최대화하는 방향으로 학습되기 때문에, 데이터 수가 많은 다수 클래스에 편향 (Bias)되는 경향이 있다. 이로 인해 모델은 소수 클래스의 중요한 특징을 학습하지 못하고 이를 노이즈로 간주하거나 무시하게 되며, 결과적으로 소수 클래스에 대한 재현율 (Recall)과 정밀도 (Precision)가 현저히 떨어지는 문제가 발생한다. 특히 3D 프린팅 적층 제조 (Additive Manufacturing)와 같은 고도의 정밀 공정에서는 결함 데이터의 확보가 매

우 어렵고 비용이 많이 들기 때문에, 이러한 데이터 희소성 (Data Scarcity) 문제는 모델의 신뢰성을 저해하는 치명적인 요인이 된다. 따라서 소수 클래스의 데이터 분포를 효과적으로 보정하여 분류 성능을 개선하는 것은 학계와 산업계 모두에서 시급한 과제이다.

이러한 클래스 불균형 문제를 완화하기 위해, 기존 연구들은 소수 클래스 데이터를 인위적으로 확장하는 다양한 data synthesis 방법을 제안해 왔다. 대표적으로 SMOTE[8], Borderline-SMOTE[10], ADASYN[11]과 같은 전통적인 oversampling methods 는 비교적 단순한 방식으로 데이터 분포를 보정할 수 있다는 장점으로 인해 널리 활용되어 왔으나, 이미지와 같이 고 차원적이고 복잡한 구조를 지닌 데이터에서는 의미론적 다양성을 충분히 반영하지 못한다는 한계가 반복적으로 지적되어 왔다. 이러한 한계를 극복하기 위한 대안으로, 최근에는 GAN[12], DCGAN[15], CGAN[13]과 같은 generative models 기반 data synthesis 가 주목받고 있으며, 이들은 데이터의 잠재 분포를 학습함으로써 기존 샘플링 방식보다 풍부한 시각적 변이를 생성할 수 있다는 점에서 이미지 도메인에서 가능성을 보여주었다. 한편, Diffusion Models[20]과 같은 최신 생성 모델들은 매우 높은 품질의 이미지 생성을 가능하게 하지만, 높은 계산 비용과 느린 샘플링 속도로 인해 실제 제조 환경에 적용하기에는 현실적인 제약이 존재한다.

이러한 다양한 data synthesis 방법들이 생성하는 이미지의 품질 차이가 실제 분류 성능에 어떠한 영향을 미치는지, 특히 적층 제조와 같이 미세한 표면 패턴이 중요한 산업 도메인에서 이에 대한 체계적인 비교 분석은 충분히 이루어지지 않았다. 본 연구는 이러한 연구 공백에 주목하여, Chung et al.[7]에서 제공된 3D 프린팅 적층 제조 표면 이미지 데이터셋을 바탕으로 전통적인 oversampling methods 와 GAN 계열 generative models 를 data synthesis 에 병행 적용

하고, 생성된 이미지의 품질과 supervised CNN classifier 의 분류 성능 개선 가능성과 관계를 현상적으로 분석함으로써 실용적인 데이터 불균형 완화 전략에 대한 시사점을 제공하고자 한다.

2. Related Work

2.1 Various Sampling Methods

현실 세계에서는 제조, 의료, 금융, 감성 분석, 생물학 등 다양한 분야에서 데이터의 불균형 문제가 발생한다. 특정 클래스에 균형이 치우쳐 있는 경우, Convolutional Neural Network (CNN)[21] 분류기가 다수 클래스 쪽으로 편향될 수 있다. 이러한 불균형 데이터 문제를 해결하는 데 있어 연구자들은 데이터 수준 혹은 알고리즘 수준에서 해결책을 제시한다[1].

Babu et al. [1] 은 sentiment analysis, medical field, intrusion detection, fraud detection 네 가지 분야의 불균형 데이터셋 증강에 두 수준의 다양한 방법론을 적용한 39 개 선행연구에 관한 요약과, 각 방법론 별 성능을 비교하였다. 아래는 각 분야에서 최고 성능을 보인 연구 및 사용된 기법이다. 11 개의 Sentiment analysis 연구에서는 undersampling 기법인 Tomek link method 를 OS log dataset 에 적용하여 99.93%의 정확도를 얻었다 [2]. 11 개의 Medical field 연구에서는 Double GAN 을 plant village dataset 에 적용하여 99.53%의 정확도를 얻었다 [3]. 11 개의 Intrusion detection 연구에서는 NSL-KDD 와 CIS-IDS2017 datasets 에 Adaptive Synthetic Sampling (ADASYN) 을 적용하여 99.91%의 정확도를 보였다 [4]. 마지막 6 개의 Fraud detection 연구에서는 transaction's dataset 에 semantic fusion of k-means cluster 를 적용하여 98.9%의 정확도를 얻었다 [5].

해당 연구에서는 Synthetic Minority Over-sampling Technique (SMOTE)을 기반으로 한 oversampling method 인 ADASYN 이 최고 성능을 보였다. 이와 대비되는 undersampling 의 대

표적인 기법으로는 Random Under-Sampling(RUS)이 존재하는데, 이때 RUS 에서는 막대한 양의 데이터가 폐기되기 때문에 중요한 정보를 포함한 데이터가 삭제될 가능성이 있다. 이러한 데이터의 손실은 소수 클래스 샘플과 다수 클래스 샘플 간의 결정 경계(borderline) 학습을 어렵게 만들어 분류 성능 저하를 초래할 수 있다 [6]. 선행 연구의 최고 성능 결과 및 undersampling 의 문제점에 기반하여 본 연구는 데이터 수준의 oversampling 기법, 즉 데이터 증강에 주목하였다.

2.2 Data Augmentation in Additive Manufacturing Processes

다른 분야와 마찬가지로, 실제 제조 공정에서는 비정상 상태가 정상 상태에 비해 극도로 작은 비율로 발생한다. 이러한 불균형 데이터셋으로 인하여 비정상 상태를 감지하는 분류기의 성능이 저하된다. Chung et al. [7] 은 제조 공정의 불균형 데이터셋 문제를 해결하기 위해 Generative Adversarial Network (GAN)[12] 기반의 새로운 데이터 증강 방법을 제안하였다.

Chung 이 제안한 Three-player structure 는 기존 generator 와 discriminator 의 경쟁적인 관계에 generator 와 classifier 사이의 상호협력적 학습 (Cooperative Learning)을 추가하여, 생성자가 더 좋은 품질 (i.e., realistic and state-distinguishable) 의 샘플을 생성할 수 있도록 하였다. Three-player 방법론의 효율은 오픈 소스 데이터인 MNIST fashion data 와 polymer additive manufacturing process data, 그리고 metal additive manufacturing process data 의 세 가지 케이스 스터디를 통해 측정되었다. 모든 케이스 스터디에 sampling-based methods (SMOTE, borderline-SMOTE, ADASYN)과 GAN-based methods (CDRAGAN, BAGAN-GP, Cooperative GAN), 마지막으로 three-player 를 각각 적용하였다. 결론적으로, 소수 클래스의 훈련 샘플 수가 늘어날 수록 모든 방법론이 baseline 에 비해 성능이 향

상되었으며, 그 중 three-player 의 성능이 가장 효과적으로 개선됨을 입증하였다.

Polymer AM process data 케이스 스터디의 경우, 두 개 비정상 클래스의 balanced ratio 를 정상 클래스 대비 0.10, 0.15, 0.20 의 비율로 설정하였다. Three-player 는 해당 케이스의 F-score, Precision, Recall 에서 각각 3%~13%, 3%~10%, 2%~10%의 성능 향상을 보였다. 본 연구진은 metal AM process data 케이스 스터디의 결과에도 주목하였는데, 해당 데이터의 성능은 4.1 MNIST data case study result(임시 소제목)에서 서술한 바와 같이, oversampling 기반 방법이 GAN 기반 방법보다 높은 성능 향상 결과를 보였다. 전통적 샘플링 기법보다 발전된 모델인 GAN 이 더 높은 성능을 보일 것이라 예측한 본 연구의 가설과는 달리, 이러한 결과가 나온 원인은 실제 이미지 수가 극도로 적고, highly overlap 되었기 때문이라고 예상된다[7].

해당 연구는 기존 oversampling 방법과 GAN 기반 방법의 한계점을 극복한 새로운 구조의 모델을 제시함으로써, 실제 적층 제조 공정 데이터의 클래스 불균형 문제를 해결하는데 기여할 것이다. 또한 적층 제조 공정 데이터뿐만 아니라 어떤 제조 데이터에도 적용이 가능하다는 점에서 의의가 있다.

2.3 Oversampling Methods

가장 잘 알려진 oversampling 기법은 Synthetic Minority Over-sampling Technique(SMOTE)으로, 2002 년 Chawla et al. [8]에 의해 처음 제안되었다. SMOTE 는 소수 클래스 인스턴스를 단순 복제하는 대신, 정의된 이웃 안에 존재하는 여러 인스턴스 사이의 보간(interpolation)을 통해 만들어진다 [9]. 먼저, 학습 데이터의 소수 클래스 인스턴스 하나를 무작위로 선택한다. 그 다음 해당 인스턴스의 k 개의 최근접 이웃(k -nearest neighbors, kNN)을 찾고 [6], 이 k 개의 인스턴스 중 n 개가 무작위로 선택되어 새로운 인스턴스 생성을 위한 보간에 사용된다.

SMOTE의 유용성이 입증된 이후, 다양한 시나리오의 데이터셋 학습 성능을 개선하기 위한 많은 확장과 대안이 제안되었다 [9].

Borderline-SMOTE은 2005년 Han et al. [10]에 의해 제안되었다. 이 알고리즘은 경계선에서 멀리 떨어진 샘플들이 분류 성능에 거의 영향을 미치지 않을 것이라는 전제에 기반한다. 이 기법은 소수 클래스 인스턴스의 이웃 내 다수 예시와 소수 예시 간의 적절한 비율로 경계선에 속하는 예시들을 식별한 후 oversampling 한다 [10]. Borderline-SMOTE은 SMOTE의 생성 공식을 그대로 사용하지만, oversampling 하는 샘플 선택에 차이가 존재한다.

2008년 He et al.에 의해 제안된 Adaptive Synthetic Sampling(ADASYN)은 소수 데이터 샘플을 분포에 따라 적응적으로 생성한다는 아이디어에 기반한다. 즉, 학습하기 쉬운 샘플보다 학습하기 어려운 샘플에 대해 더 많은 합성 데이터를 생성함으로써, 결정 경계를 적응적으로 이동시킨다 [11]. 이때 소수 클래스 샘플 주변의 kNN 중 다수 클래스 비율이 높을수록, 그 소수 클래스 샘플은 학습하기 어려운 샘플(hard-to-learn)로 간주된다.

2.4 Introduction of Generative Models

Data synthesis 관점에서 Generative Adversarial Networks(GAN) [12]은 복잡한 고차원 분포의 데이터를 잘 모사하는 생성 모델을 구현하고, 그 모델로부터 고품질의 샘플을 생성하는 데 목적이 있다. 이후 GAN의 구조를 변형 및 확장한 CGAN[13], CycleGAN[14], DCGAN[15], WGAN[16], WGAN-GP[17], LSGAN[18], StyleGAN[19] 등의 후속 모델들이 제안되었다. GAN 계열의 생성형 모델은 이러한 발전과정을 이루어 왔으며, 최근에는 Diffusion[20] 등 다른 형태의 생성형 모델에 대한 연구도 활발히 이루어지고 있다.

우리는 이러한 연구 흐름 속에서 적층 제조 데이터 선행연구[7]의 아이디어에 중점을 두고

생성형 모델 중 GAN 계열을 방법론으로 채택하였다. 그러나 GAN은 학습의 불안정성과 종종 생성자의 무의미한 출력(i.e., 노이즈 벡터에 가까운 샘플 또는 생성 이미지의 다양성 확보가 어려움)을 발생시키기도 한다[15]. 본 연구에서는 GAN의 이런 근본적인 한계점을 적층 제조 도메인에서 현상적으로 마주하며, 이를 해결할 수 있는 GAN 기반의 확장형 모델인 Conditional Generative Adversarial Nets(CGAN), Deep Convolutional Generative Adversarial Networks(DCGAN) 모델까지 적용하고자 한다. 자세한 방법론에 대해서는 이어지는 Sec. 3에서 다루도록 하겠다.

3. Method

본 연구는 Chung et al.[7]에서 제안된 접근법을 바탕으로, 적층 제조(Additive Manufacturing) 도메인에서 발생하는 심각한 클래스 불균형 문제를 해결하기 위해 GAN 기반 data synthesis를 적용하였다. 이후 전통적인 oversampling methods 대비 supervised classifier의 분류성능을 얼마나 향상시키는지 그 효과를 분석하고 생성된 이미지 품질을 심층적으로 분석하였다.

이를 위해 먼저 실제 공정에서 관측되는 데이터의 분포를 반영하여[7] 불균형 학습데이터셋을 구축하고, 해당 데이터를 이용해 CNN 기반 classifier의 baseline 성능을 산출하였다. 이후 oversampling methods와 generative models를 사용하여 소수 클래스의 이미지를 정상 클래스와 동일한 수준까지 확장하여 균형 학습데이터셋을 재구성하였다. 이러한 균형 데이터셋을 기반으로 동일한 구조의 CNN classifier를 초기화하여 재학습을 수행하였고, 학습 전후의 성능 변화를 정량적으로 비교하였다.

더불어 생성된 이미지의 품질을 정성적으로 분석하여, GAN이 소수 학습데이터로 클래스의 특징을 어느 정도 재현하는지 면밀히 검토하였다. 또한 GAN의 확장 모델인 DCGAN(Sec. 4.3), CGAN(Supp. Sec. B)을 추가적으로 실험하여,

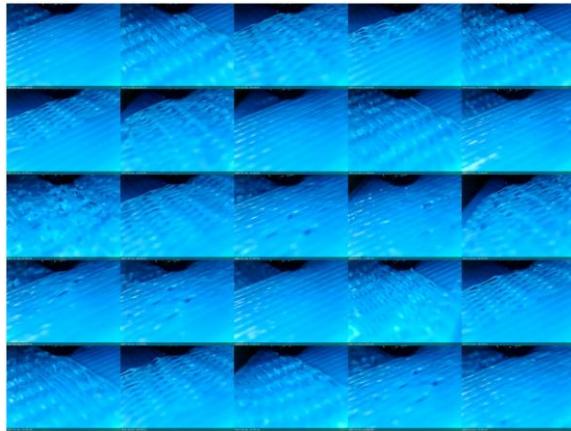


Figure 1. Type1(Underfill_FR50) TrainSet Images

Table 1. AM Dataset Pre-processing

	Given	+Augmentation	Train	Test
Normal	305	610	500	100
FR50(type1)	197	394	25	100
Fan(type2)	153	306	25	100

GAN 이 갖는 한계점과 그 구조적 개선이 가져오는 이미지 품질 향상 및 데이터 합성의 효과성을 현상적으로 파악하고자 하였다.

3.1 AM Dataset Pre-processing

본 연구에서 사용된 데이터는 3D 프린팅 공정 (Fused Filament Fabrication)에서 추출된 고해상도 이미지로, 정상(normal) 클래스 외에 필라멘트 공급 부족으로 인해 발생하는 Underfill_FR (type1)과 냉각 팬의 과도한 작동으로 인해 발생하는 Underfill_Fan (type2) 두 종류의 비정상 상태 이미지를 포함한다. Underfill_FR 은 재료 공급량 부족으로 인해 적층 라인의 간격이 넓어지며 표면 밀도가 저하되는 형태로 나타나고, Underfill_Fan 은 과도한 냉각으로 적층 패턴이 미세하게 단절되거나 균열이 발생하는 모습으로 나타난다[7].

전체 보유 데이터는 normal 305 장, type1 197 장, type2 153 장으로 구성되어 있으며, 실제 제조공정의 불균형 특성을 반영하기 위해 소수 클래스 비율을 정상 대비 0.1 이하 수준으로 유지하고자 하였다. 또 좌우반전 augmentation 을 추가적으로 적용하여 데이터 다양성을 확보하고자 하였다. 최종적으로 Table 1 과 같이 학습데

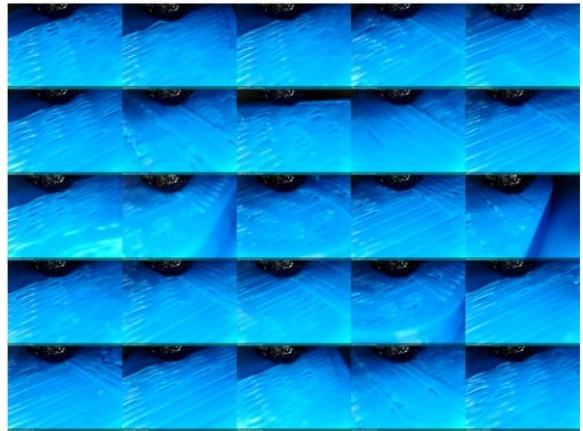


Figure 2. Type2(Underfill_Fan) TrainSet Images

이터를 구성하였으며, 테스트 데이터셋의 경우 학습데이터셋을 구성하고 남은 데이터를 활용하여 각 클래스별 100 장으로 구성하였다. 비정상 각 클래스의 학습데이터셋에 사용된 이미지 데이터는 Figure 1 과 Figure 2 에서 확인할 수 있다.

3.2 CNN Classifier

Convolutional Neural Network(CNN)은 이미지와 같은 격자 구조 데이터로부터 지역적 특징을 효과적으로 추출하기 위해 설계된 신경망 구조로, 합성곱 연산을 기반으로 입력 이미지의 공간적 패턴을 계층적으로 학습한다. 초기 층에서는 가장자리, 코너와 같은 저수준 특징을 포착하며, 층이 깊어질수록 점차 더 추상적이고 복잡한 형태의 고수준 특징을 학습한다. 합성곱 계층(convolution layer)은 필터를 사용하여 지역적 정보만을 고려한 특징 맵을 생성하고, 폴링(pooling) 계층은 공간적 차원을 축소하여 중요한 정보는 보존하면서 연산량을 감소시키는 역할을 한다. 이러한 구조적 특성 덕분에 CNN 은 완전연결 신경망에 비해 파라미터 수가 크게 줄어들 뿐만 아니라, 이미지의 공간적 불변성(spatial invariance)을 자연스럽게 학습할 수 있어 이미지 분류에 널리 사용된다.[21]

본 연구에서 성능 향상을 목표로 하는 supervised classifier 역시 CNN 구조를 기반으로 구현하였다. 우리의 AM 학습데이터셋은

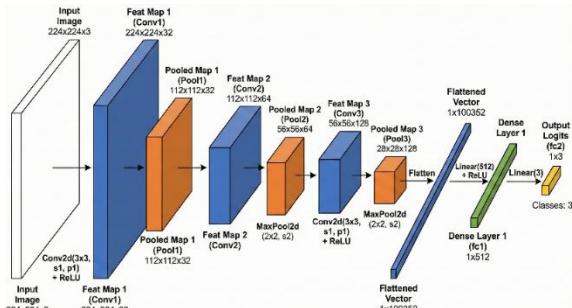


Figure 3. Our CNN Classifier Architecture

normal, type1, type2 3 개 클래스로 구성되므로 multi-class CNN classifier 구조로 구현되었으며 3 개의 layer, 2 개의 FC layer 로 구성되었다. 구체적인 구조는 Figure 3 에서 확인할 수 있다.

3.3 Generative Models

Generative Adversarial Network

Generative Adversarial Network (GAN)은 실제 데이터 분포를 직접 모사하여 새로운 샘플을 생성하는데 매우 효과적인 구조를 갖는다. GAN은 생성자(Generator, G)와 판별자(Discriminator, D)라는 두 신경망이 서로 경쟁적으로 학습된다는 점에서 기존의 likelihood 기반 생성 모델들과 구별된다. 생성자는 잠재 공간에서 샘플링된 벡터를 이용해 실제 데이터와 유사한 이미지를 생성하도록 학습되고, 판별자는 입력된 이미지가 실제 데이터에서 온 것인지 혹은 생성자가 만든 것인지 판별하는 이진 분류기 역할을 수행한다. 두 네트워크는 상반된 목적을 갖고 학습을 수행하며, 이러한 적대적 관계를 통해 생성자는 점차 더 정교한 데이터를 생성할 수 있도록 개선된다.[12]

GAN의 학습 목적은 생성자와 판별자 사이의 미니맥스(minimax) 최적화 문제로 정의되며, 이는 Eq. 1과 같은 목적함수로 표현된다. Eq. 1에서 판별자는 실제 데이터에 대해 높은 확률을 부여하고 생성된 데이터에 대해서는 낮은 확률을 부여하도록 학습되며, 반대로 생성자는 판별자가 생성된 이미지를 실제로 오인하도록 만드는 방향으로 학습된다. 이러한 구조는 두 네트워크를 각각 향상시키는 경쟁적 관계를 유도

하며, 결과적으로 생성자가 학습 데이터의 분포를 점차 정확하게 모사하게 한다. 고정된 생성자 G에 대해 최적의 판별자 D는 Eq. 2와 같이 유도된다. Eq. 2를 Eq. 1에 대입하여 단순화하면, 생성자 학습의 목적함수는 Jensen-Shannon Divergence (JSD)를 기반으로 Eq. 3과 같은 형태로 표현된다. 이때 JSD는 두 분포가 완전히 동일할 때에만 0이 되며, 생성자 분포가 실제 데이터 분포와 일치할 때 전체 목적함수는 최솟값 $-\log 4$ 에 도달한다. 이는 생성자가 완전히 현실적인 데이터를 생성하여 판별자가 실제와 생성 여부를 구분하지 못하는 상태에 해당한다.

$$\min_G \max_D V(D, G) = E_{x \sim p_{dt}(x)} [\log D(x)] + E_{z \sim p_g(z)} [\log (1 - D(G(z)))] \quad (1)$$

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad (2)$$

$$C(G) = -\log 4 + 2 \cdot \text{JSD}(p_{data} \| p_g) \quad (3)$$

그러나 Eq. 1 의 구조는 실제 학습 과정에서 여러 한계를 가진다. 대표적으로 vanishing gradient 문제가 발생하는데, 이는 학습 초기 생성자가 매우 낮은 품질의 이미지를 생성할 경우 판별자가 $D(G(z)) \approx 0$ 을 출력하게 되고, 이 때 생성자의 손실함수의 기울기 $\log(1-D(G(z)))$ 는 거의 0에 수렴하여 생성자에게 의미 있는 학습 신호가 전달되지 않는다. 또 다른 주요 문제는 mode collapse 로, 생성자가 판별자를 속일 수 있는 소수의 패턴만 반복적으로 생성하며 데이터 분포의 다양성을 포착하지 못하는 현상이다. 이러한 문제는 생성자와 판별자의 학습 균형이 쉽게 깨지고, 그 결과 학습이 불안정해지기 쉬운 GAN 의 구조적 특성에서 비롯된다.

본 연구에서는 데이터 불균형 문제를 해결하기 위한 data synthesis 수단으로 GAN을 적용하였다. 이를 위해 잠재 벡터 z를 다층 퍼셉트론 기반의 생성자 네트워크에 입력하여 이미지 공간으로 매핑하고, 판별자는 flatten된 이미지 벡터를 입력받아 실제 여부를 판별하도록 구성하

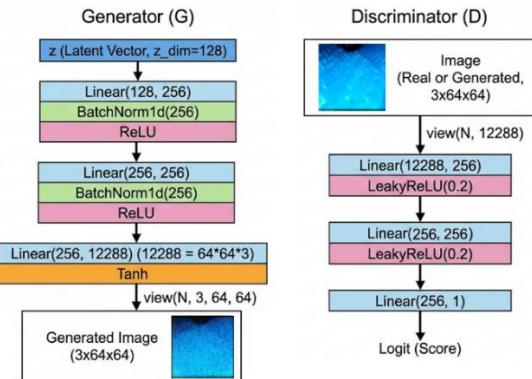


Figure 4. Our GAN Architecture for AM Data Synthesis

였다. Figure 4에 제시된 본 연구의 GAN 구조는 비교적 단순한 fully-connected 기반 아키텍처로 구성되었으며, Batch Normalization과 ReLU 활성화를 적용하여 훈련의 안정성을 확보하고, 최종 출력층에서 Tanh 함수를 사용해 이미지 픽셀 범위로 매핑하였다. 판별자 역시 LeakyReLU 기반의 다중 선형 계층으로 구성되어, 생성 이미지와 실제 이미지를 구분하는 판별 점수를 최종적으로 출력하는 구조를 갖는다.

Deep Convolution Generative Adversarial Network

Deep Convolutional Generative Adversarial Networks (DCGAN)은 기존 GAN의 불안정한 학습 특성과 제한적인 표현력을 개선하기 위해 합성곱 신경망 구조를 도입한 확장형 GAN이다. 기존 GAN은 다층 퍼셉트론 기반의 완전연결층을 사용함으로써 이미지의 공간적 구조를 직접적으로 반영하기 어렵고, 특히 고해상도 이미지 생성 시 공간적 패턴 손실이 쉽게 발생하는 한계가 있었다. 이에 반해 DCGAN은 합성곱과 전치 합성곱을 기반으로 한 계층적 구조를 사용하여 이미지의 지역적·전역적 패턴을 효과적으로 학습하며, GAN 대비 학습 안정성과 생성 이미지의 시각적 품질을 크게 향상시킨다.[15]

DCGAN의 핵심적 설계 원리는 안정적인 생성 모델 학습을 위한 구조적 가이드라인에 기반한다. 첫째, 풀링(pooling) 연산을 사용하지 않고 생성자에서는 분수 스트라이드 합성곱을

활용하여 점진적으로 공간 해상도를 증가시키며, 판별자에서는 스트라이드 합성곱을 사용해 공간 정보를 압축한다. 이를 통해 네트워크가 이미지의 공간 변환을 직접 학습할 수 있도록 하여, 생성 품질의 세밀도를 높인다. 둘째, 전통적인 완전연결층을 대부분 제거하고, 생성자의 입력 노이즈 벡터를 먼저 고차원 텐서로 변환한 뒤 합성곱 스택을 통해 이미지를 구성하는 방식을 사용한다. 이러한 구조는 불필요한 파라미터 수를 줄이고 학습 안정성을 높인다. 셋째, 생성자와 판별자 모두에 batch normalization을 도입하여 기울기 흐름을 원활하게 하고, 학습 초기의 모드 붕괴 가능성을 감소시키는 방향으로 작용한다. 마지막으로 생성자에서는 ReLU를 기본 활성화 함수로 사용하고, 최종 출력층에서 Tanh를 적용하여 이미지 픽셀 값의 범위를 적절히 관리하는 구조를 채택하며, 판별자에서는 LeakyReLU를 사용하여 음수 축에서의 기울기 소실을 완화한다.

DCGAN은 기존 GAN 대비 구조적 개선을 통해 안정적인 학습을 가능하게 하고, 이미지의 공간적 패턴을 정교하게 포착함으로써 생성 이미지의 품질을 높인다는 장점을 지닌다. 또한 판별자가 학습한 합성곱 기반 표현은 지도 학습 과제에 전이 학습으로 활용될 수 있으며, 생성자의 잠재 공간에서도 벡터 연산을 통해 의미 있는 시각적 변형이 관찰되는 등 표현 학습 측면에서 여러 응용 가능성을 제공한다. 그럼에도 DCGAN은 여전히 mode collapse와 같은 GAN 특유의 구조적 한계를 완전히 극복하지는 못한다.

본 연구에서 사용한 DCGAN의 전체 아키텍처는 Figure 5에 제시되어 있다. 생성자는 잠재 벡터를 입력받아 전치 합성곱과 Batch Normalization, ReLU 기반의 계층을 거쳐 점진적으로 해상도를 확장하며 최종적으로 $3 \times 128 \times 128$ 크기의 이미지를 출력한다. 판별자는 입력 이미지로부터 합성곱과 LeakyReLU 기반의 계층을 거치면서 특징을 추출하고, 마지-

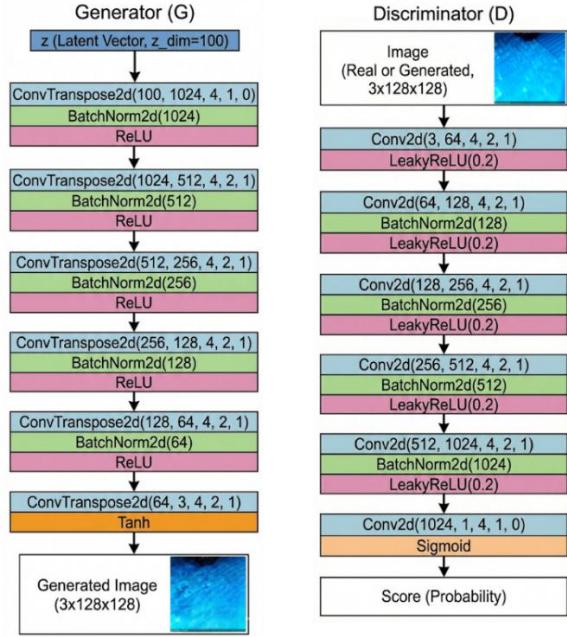


Figure 5. Our DCGAN Architecture for AM Data Synthesis

막으로 sigmoid 함수를 사용하여 입력 이미지가 실제 데이터인지 생성된 데이터인지에 대한 확률 값을 산출한다.

Conditional Generative Adversarial Network

Conditional Generative Adversarial Network (CGAN)은 GAN의 확장 구조로, 생성 과정에 조건 정보를 명시적으로 부여함으로써 출력 데이터의 모드를 제어할 수 있도록 설계된 조건부 생성 모델이다. 일반적인 GAN이 데이터 분포 전체를 비지도 방식으로 모사하는 데 그치는 반면, CGAN은 클래스 레이블과 같은 외부 조건 y 를 함께 입력함으로써 특정 클래스의 데이터를 선택적으로 생성할 수 있다는 점에서 보다 제어된 방식의 데이터 합성이 가능하다. 이러한 구조는 생성자에게는 z 와 y 의 결합 입력을 제공하고, 판별자 역시 실제 혹은 생성된 샘플과 조건 정보를 함께 입력받아 학습한다.[13] 이에 따라 CGAN의 목적 함수는 기존 GAN의 목적 함수에 조건 정보를 추가한 형태로 Eq. 4와 같이 정의된다.

$$\min_G \max_D V(D, G) \\ = E_{x \sim p_{dt}(x)}[\log D(x | y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z | y)))] \quad (4)$$

Eq. 4에서 $D(x | y)$ 는 조건 y 가 주어졌을 때 입

력 x 가 실제 데이터일 확률을 나타내며, $G(z | y)$ 는 동일한 조건 y 하에서 생성된 샘플을 의미한다. 조건을 명시적으로 부여함으로써 생성자는 특정 클래스의 특징을 보다 안정적으로 학습하도록 유도되며, 이는 모드 붕괴를 완화하고 다양한 데이터 모드를 보다 효과적으로 학습할 수 있다는 장점을 제공한다.

GAN은 데이터 분포를 비지도 방식으로 학습하므로, 본 연구와 같이 소수 클래스의 데이터가 극단적으로 부족한 환경에서는 DCGAN 방식도 유의미한 특징을 포착하지 못하거나 학습이 붕괴될 위험이 크다. 이러한 구조적 한계를 보완하기 위해, 본 방법론에서는 class label을 생성 및 판별 과정의 조건으로 활용하는 CGAN을 future work로 제안한다. 이는 제한된 샘플 내에서도 클래스 고유 정보를 명시적으로 제공함으로써, 소수 클래스에 대한 생성 안정성을 확보하기 위한 필수적인 조치이다. 본 연구에서는 AM 데이터셋에는 CGAN을 적용하지 못했지만, **Supp. Sec. B**에서 MNIST 소수 클래스 데이터가 매우 극단적인 상황에서 CGAN의 학습 양상을 통해 그 가능성을 파악하고 future work로서 방법을 제안하게 되었다.

4. Experiment

우리는 **Sec 3.1**에서 구현한 학습데이터셋을 기반으로 **Sec 3.2**의 generative models를 사용하여 data synthesis를 진행한 후, **Sec 3.2**에서 구현한 CNN classifier의 분류성능을 재평가하였다. 그리고 실제 성능향상에 미치는 영향을 전통적인 oversampling methods를 적용한 결과와 비교분석하고자 하였다.

4.1 AM Data Synthesis by Oversampling Methods

Evaluation Compared to Baseline. Oversampling Methods로는 SMOTE, B-SMOTE, ADASYN을 선택하였고, 이를 학습데이터셋에 각각 적용하여 균형 데이터셋을 구성하였다. 이후 동일한 구

Table 2. Evaluation Results per Data Synthesis Methods

Method	Precision	Recall	F1-score
Baseline	0.8342	0.7467	0.7369
SMOTE	0.8561	0.7467	0.7206
B-SMOTE	0.8593	0.7567	0.7417
ADASYN	0.8619	0.8000	0.7797
DCGAN	0.9048	0.8667	0.8611

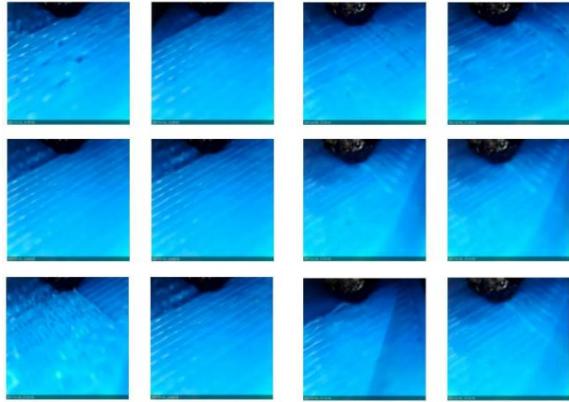


Figure 6. Synthesized Type 1 (left) & 2 (right) Images by SMOTE (Top), B-SMOTE (Middle), and ADASYN (Bottom)

조의 CNN을 재학습하여 precision, recall, f1-score 중심으로 baseline 대비 성능 변화를 비교하였다. 각 기법이 적용되었을 때의 성능 변화는 Table 2에 요약하였다.

Visual Analysis. Oversampling methods를 통해 합성된 이미지들은 Figure 6에서 확인할 수 있다. Oversampling Methods를 통한 Data Synthesis는 이미지 레벨에서의 interpolation 연산을 통해 산출된 이미지들이기 때문에 기준이 되는 이미지에 다른 이미지들의 잔상이 겹쳐진 듯한 이미지들을 합성해낸다. 그러나 AM 학습데이터셋 25장을 시각적으로 확인해보면, 다양한 각도, 다양한 위치에서 프린팅되는 실시간 이미지는 점에서 단순 이미지 레벨에서의 interpolation을 적용하게 되면 본래 비정상 특징이 나타난 표면에서 정상 특징이 겹쳐지게 된다. 이에 따라 각 비정상 유형에서 핵심적인 차이가 될 수 있는 색상, 명암 등의 특징을 명확히 잡아내는 데 큰 한계가 있음을 알 수 있다.

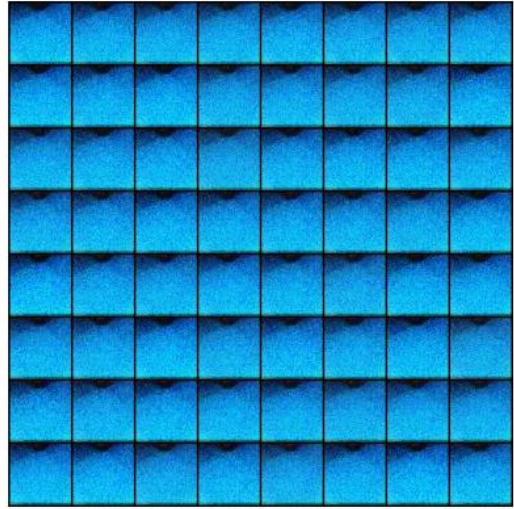


Figure 7. Synthesized Type 1 Images by GAN

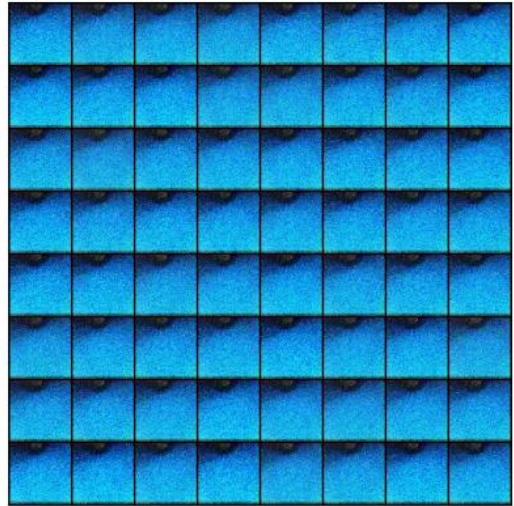


Figure 8. Synthesized Type 2 Images by GAN

4.2 AM Data Synthesis by GAN

Visual Analysis. GAN을 통해 합성된 이미지들은 Figure 7, 8에서 확인할 수 있다. 학습 도중 지속적으로 G 쪽에 유의미한 기울기 신호를 형성하는 것을 확인하여 600 epoch까지 진행하였으나, GAN 합성 결과는 oversampling methods 보다 훨씬 더 noisy함을 확인할 수 있으며, AM 이미지 자체가 가지고 있는 일반적인 특징조차 세밀하게 잡아내지 못함을 알 수 있었다. 또 생성되는 이미지가 대체로 일관된다는 mode collapse 현상도 발견할 수 있었다. 특히 이는 잠재공간 dimension, D와 G의 learning rate 등의 hyperparameter tuning을 통해서도 해결할 수 없

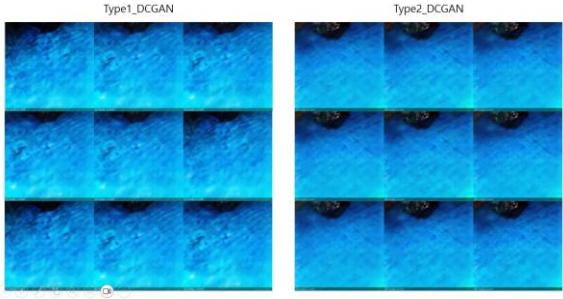


Figure 9. Synthesized Type 1 (left) & 2 (right) Images by DCGAN

는 문제였다. 우리는 GAN이 합성하는 이미지의 품질이 이렇게 떨어지는 이유를 적은 학습 데이터수 때문도 있지만 고화질 이미지 데이터셋을 단순 FC layer로 처리하는 과정에서 세밀한 특징들을 잡아내는 데 한계가 있기 때문으로 보았다. 매우 noisy한 합성 이미지의 품질과 mode collapse 현상 발견에 따라 별도의 CNN classifier evaluation을 진행하지 않았고, Sec. 4.3에서 resolution 문제를 개선할 수 있는 DCGAN을 통해 실험 과정을 재진행하였다.

4.3 AM Data Synthesis by DCGAN

Evaluation Compared to Baseline. 앞선 Sec. 4.2에서 발견한 이미지 품질 문제를 개선하기 위해 DCGAN을 사용하여 data synthesis 과정을 재진행하였다. 이후 동일한 구조의 CNN을 재학습하여 precision, recall, f1-score 중심으로 baseline 대비 성능 변화를 확인하였고, 이를 Table 2에 요약하였다. DCGAN 합성을 통해 합성된 이미지로 균형데이터를 형성한 후 CNN을 재학습시켰을 때 precision 0.0706, recall 0.12, F1-score 0.1242 향상되었다. 또한 Supp. Sec. A의 Figure 10, 11, 12, 13, 14의 confusion matrix를 보면 oversampling method를 적용하였을 때 baseline 대비 type1에 대한 구분력이 오히려 하락하는 결과를 보였으나, DCGAN을 적용하였을 때는 type1 테스트데이터셋 100장 중 60장을 정확히 분류해내어, 이 부분이 핵심적으로 개선된 모습을 볼 수 있었다. 이는 oversampling methods를 적용했을 때의 성능개선 대비 더 효과적이고 유의미한 개선이 이루어졌음을 의미한다.

Visual Analysis. DCGAN을 통해 합성된 이미지들은 Figure 9에서 확인할 수 있다. 훈련은 500 epoch까지 진행하였고, noisy하여 특징을 확인하기 어려웠던 GAN 합성 이미지들에 비해 각 비정상 유형들이 나타내는 핵심적인 특징들은 표현이 되고 있음을 시각적으로 확인할 수 있었다. 특히 type1에서 재료부족으로 인해 출력된 표면 위의 재료 간 공백이 생기면서 색의 명암차가 생기고 경계가 구분된다. 이런 경계부분을 합성 데이터가 직선과 곡선 등으로 정교하게 표현하지는 못하지만 색감, 명암 차이는 명확하게 포함하고 있는 이미지들을 합성하고 있음을 확인할 수 있었다. 또 type2의 경우 type1에서 드러나는 색감, 명암 차이보다는 재료가 적절한 온도를 받지 못하면서 곡선형 또는 입체적인 재료의 출력이 나타난다는 것이 특징이다. 마찬가지로 곡선 등 정교한 표현력으로 합성하지는 못하였지만 직선으로 평행하고 일관되게 출력되어야 하는 표면의 상태를 blurry하게 표현하고 있음을 알 수 있었다. Oversampling methods를 통해 합성된 이미지들과 비교해보면, 이미지의 해상도 측면에서는 약점을 보이지만, 단순 interpolation을 통해 정상과 비정상 구분이 모호한 특징들을 가진 이미지들을 합성하는 것보다 특정 비정상 유형이 가지는 특징을 핵심적으로 합성하는 것이 분류 성능 향상에 더 효과적임을 알 수 있었다.

Limitations. DCGAN 합성 이미지들을 자세히 보면, 각 비정상 유형의 핵심적인 특징을 표현하고 있지만 그 특징을 다양하게 생성해내지는 못하고 있다는 것을 확인할 수 있다. 이는 DCGAN을 사용하면서 GAN 대비 품질이 개선된 이미지를 생성할 수 있게 되었지만 생성되는 이미지의 다양성 관점에서는 개선되지 않았음을 알 수 있다. 이는 단일 클래스 내에서 이미지 합성을 진행했지만 단일 클래스 내에서도 다양한 모습과 형태로 존재하기 때문에 전형적인 GAN이 갖는 mode collapse 한계점을 DCGAN이 완벽하게 해결할 수 없다는 것을

현상적으로 확인할 수 있는 대목이다. 이에 따라 우리는 본 연구의 future work로서 conditional DCGAN[23], WGAN[16] 등 이미지 품질과 생성 이미지의 다양성 관점에서 모두 개선 가능성이 있는 generative models를 사용할 것을 제안한다.

5. Conclusion

본 연구는 적층 제조 결함 탐지에서 발생하는 클래스 불균형 문제를 해결하기 위해 전통적인 oversampling methods (SMOTE, Borderline-SMOTE, ADASYN)과 generative models (GAN, DCGAN)을 사용하고 분류 성능 향상 관점에서 비교 분석 하였다. 극단적인 소수 클래스 비율을 설계한 적층 제조 데이터셋으로 정량적 분류 성능과 정성적 이미지 품질을 종합적으로 평가하였다. 실험 결과, 다음과 같은 핵심적인 발견을 도출할 수 있었다. 첫째, GAN은 극단적인 데이터 희소성 환경에서 의미 있는 이미지를 생성하는데 실패하였으며, 600 epoch 의 확장된 학습과 하이퍼파라미터 튜닝에도 불구하고 노이즈 이미지만을 출력하였다. 이러한 한계는 학습 데이터 수와 무관하게 지속되었으며, 이는 GAN의 FC layer 기반 구조가 고해상도 적층 제조 이미지의 핵심적 특징을 포착하는 데 본질적으로 부적합함을 시사한다. 둘째, 전통적인 oversampling methods 는 분류 성능에서 중간 수준의 개선을 보였으나, 이미지 레벨에서의 interpolation 으로 인해 합성된 이미지에서 특징 중첩이 발생하여 클래스 판별적 특성을 보존하는 데 한계를 나타냈다. 셋째, DCGAN 은 모든 평가 지표에서 가장 우수한 성능을 달성하였다. 이는 oversampling methods 대비 baseline 과의 유의미한 격차를 발생시켰다. DCGAN 의 합성곱 기반 구조는 mode collapse 문제로 인한 생성 이미지의 다양성 제한에도 불구하고, type 1 (Underfill_FR)의 색상 변화 및 표면 불연속성, type 2 (Underfill_Fan)의 흐릿한 표면 패턴과 같은 핵심적 결함 특성을 성공적으로 포착하였다.

Supp. Sec. B 에서 보조 실험으로 수행한 MNIST case study 는 GAN 기반 방법의 데이터 희소성에 대한 민감성을 추가로 확인시켜 주었다. 소수 클래스당 6 장의 샘플만이 존재하는 환경(불균형 비율 0.001)에서 GAN 과 CGAN 모두 의미 있는 표현을 학습하는 데 실패한 반면, 전통적인 oversampling methods 는 여전히 일관된 성능 향상을 제공하였다.

이러한 실험 결과를 바탕으로, 제조 결함 탐지를 위한 data synthesis 에 대해 다음과 같은 실용적 가이드라인을 제시한다. 첫째, 소수 클래스 샘플이 극도로 희소한 경우(10 개 미만), 전통적인 oversampling methods 가 최소한의 계산 비용으로 불균형문제를 부분적으로 해소하고 신뢰할 수 있는 기준선을 제공한다. (2) 비교적 샘플 수가 확보된 경우(약 25 개 이상), DCGAN 이 색감, 명암과 같은 결함 특이적 특징 학습을 통해 우수한 성능을 제공한다. (3) 학습 데이터 양과 무관하게, 고해상도 이미지 합성을 위해서는 DCGAN 과 생성 모델의 구조적 개선(예: 합성곱 계층)이 필수적이다.

향후 연구에서는 DCGAN에서 관찰된 mode collapse 한계를 해결하기 위해 conditional DCGAN 및 WGAN과 같은 조건부 생성 모델 또는 이미지 품질 관점에서 개선을 가져오는 발전된 모델에 대해 탐구해야 한다. 또한, 계산 비용과 도메인 적합성에 대한 추가적인 실험과 증명이 필요하지만, 다른 도메인에서 입증된 안정성과 생성 품질을 감안할 때 Diffusion Model과 같은 최신 방향성을 갖고 data synthesis를 진행하는 것도 유망한 연구 방향으로 제시된다.

References

- [1] Kunda, S. B., & Yamarthi, N. R. (2023). A Study on Imbalanced Data Classification for Various Applications. *Revue d'Intelligence Artificielle*, 37(2), 517-524. <https://doi.org/10.18280/ria.370229>
- [2] Studiawan, H., Sohel, F., Payne, C. (2020). Anomaly detection in operating system logs with

- deep learning based sentiment analysis. *IEEE Transactions on Dependable and Secure Computing*, 18(5): 2136-2148. <https://doi.org/10.1109/TDSC.2020.3037903>
- [3] Zhao, Y., Chen, Z., Gao, X., Song, W., Xiong, Q., Hu, J., Zhang, Z. (2021). Plant disease detection using generated leaves based on Double GAN. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(3): 1817-1826. <https://doi.org/10.1109/tcbb.2021.3056683>
- [4] Liu, C., Gu, Z., Wang, J. (2021). A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning. *IEEE Access*, 9: 75729-75740. <https://doi.org/10.1109/ACCESS.2021.3082147>
- [5] Darwish, S.M. (2020). A bio-inspired credit card fraud detection model based on user behavior analysis suitable for business management in electronic banking. *Journal of Ambient Intelligence and Humanized Computing*, 11(11): 4873-4887.
- [6] Hoens, T. & Chawla, N. (2013). Imbalanced Datasets: From Sampling to Classifiers. [10.1002/9781118646106.ch3](https://doi.org/10.1002/9781118646106.ch3).
- [7] Chung, J., Shen, B. & Kong, Z.J. Anomaly detection in additive manufacturing processes using supervised classification with imbalanced sensor data based on generative adversarial network. *J Intell Manuf* 35, 2387–2406 (2024). <https://doi.org/10.1007/s10845-023-02163-8>
- [8] Chawla, N. V., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- [9] Fernández, A., García, S., Herrera, F. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, 61, 863-905.
- [10] Han, H., Wang, WY., Mao, BH. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science*, vol 3644. https://doi.org/10.1007/11538059_91
- [11] Haibo He, Yang Bai, E. A. Garcia and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. (2008). *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.
- [12] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- [13] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- [14] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).
- [15] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv 2015. arXiv preprint arXiv:1511.06434*, 5.
- [16] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.
- [17] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.
- [18] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2794-2802).
- [19] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4401-4410).
- [20] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33, 6840-6851.
- [21] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [22] LeCun, Y. (1998). The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- [23] Vasamsetti, S., Chemboli, V., Shreyas, G. S. S., & Thota, S. (2024). Conditional DCGAN for targeted generation of MNIST handwritten digits. In F. M. Lin, A. Patel, N. Kesswani, & B. Sambana (Eds.), *Accelerating discoveries in data science and artificial intelligence I: ICDSAI 2023*, LIET Vizianagaram, India, April 24–25 (pp. 235–245). Springer.

Supplementary

A. More Results and Failure Cases

Detailed Evaluation Results. 각 data synthesis 적용 이후 baseline 대비 CNN classifier의 분류성능 향상에 대한 구체적인 지표를 confusion matrix로 Figure 10, 11, 12, 13, 14에 나타내었다. Baseline의 confusion matrix를 보면 본래 normal과 type2에 대한 분류성능은 데이터셋의 불균형을 감안하여도 어느 정도 확보됨을 알 수 있었다. 다시 말해 본 연구는 type1 클래스의 normal 클래스와의 구분 능력을 향상시키는 것이 핵심적이라고 볼 수 있었다.

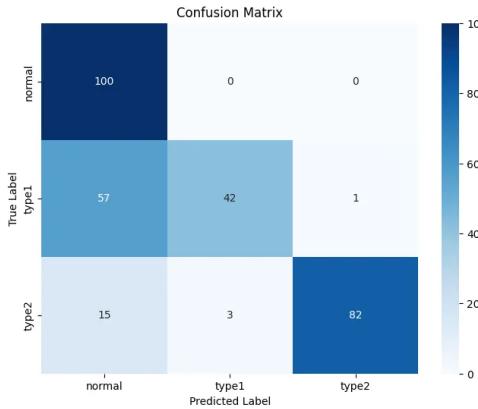


Figure 10. Confusion Matrix of Baseline Classification Performance (F1-score: 0.7369)

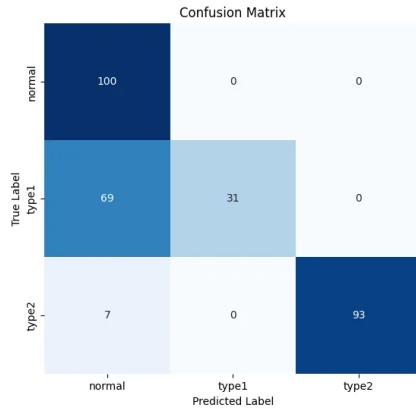


Figure 11. Confusion Matrix of After-SMOTE Classification Performance (F1-score: 0.7206)

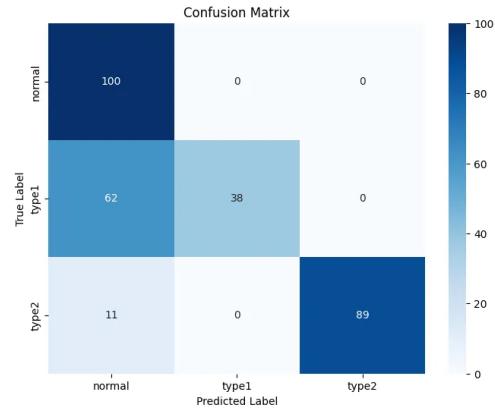


Figure 12. Confusion Matrix of After-B-SMOTE Classification Performance (F1-score: 0.7417)

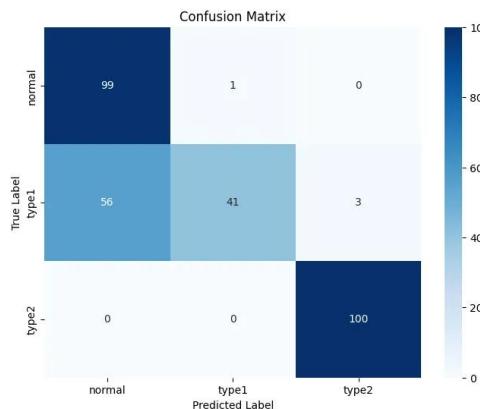


Figure 13. Confusion Matrix of After-ADASYN Classification Performance (F1-score: 0.7797)

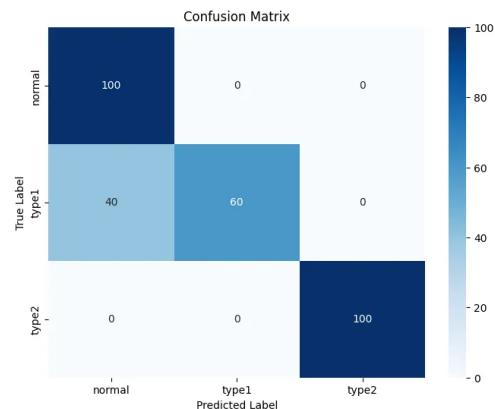


Figure 14. Confusion Matrix of After-DCGAN Classification Performance (F1-score: 0.8611)

GAN AM Full-shot Training. GAN을 통해 data synthesis를 적용할 때 지속적으로 noisy한 이미지들이 생성된다는 것이 핵심적인 문제였다. 우리는 이 문제가 단순 학습데이터수 부족 때문에 발생하는 문제인지, 혹은 GAN의 구조적 한계(FC layer로 이미지의 표현력 학습) 때문에 발생하는 문제인지 명확히 하고자 본래 pre-processing 단계 이전에 보유했던 raw 데이터셋 type1과 type2 전체를 각각 full-shot (Type1 197장, Type2 153장) 학습데이터셋으로 구성하여 600 epoch 환경에서 GAN을 다시 학습시켰다. 아래 Figure 15 결과를 통해 알 수 있듯이, 기존 극소수 학습데이터셋으로 GAN을 학습했을 때 발생한 noisy 이미지 생성문제가 full-shot으로 진행하여도 전혀 개선되지 않음을 확인할 수 있었다. 이를 통해 우리는 이미지 품질 관점에서 개선을 이루려면 학습데이터수 확보도 중요하지만 DCGAN과 같은 개선된 모델 사용이 필수적임을 알 수 있었다.

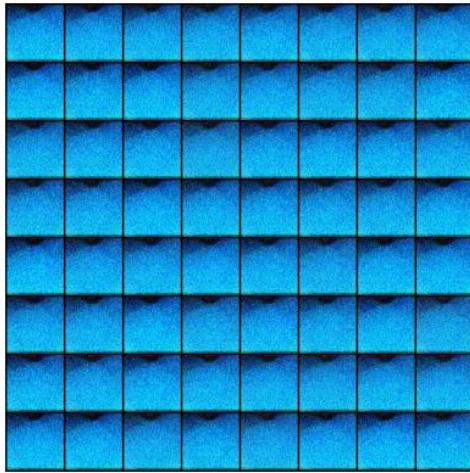


Figure 15. Synthesized Type 1 Images by GAN with Full-shot TrainSet

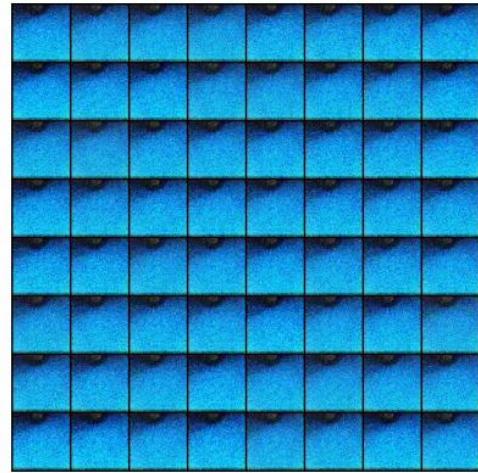


Figure 16. Synthesized Type 2 Images by GAN with Full-shot TrainSet

DCGAN Vanishing Gradient. DCGAN 학습과정에서도 특정 hyperparameter 환경에서는 vanishing gradient 현상이 발견되었다. 이는 Sec. C의 MNIST GAN hyperparameter sensitivity 실험에서 발견했던 전형적인 GAN이 가지고 있는 한계점으로, 학습초기 D의 판별력이 너무 강하고 G의 생성력이 너무 약해 유의미한 학습 기울기 형성이 이루어지지 않는 모습을 loss 값을 통해 확인할 수 있었다. 500 epoch가량 학습을 진행했음에도 전혀 개선이 이루어지지 않는 것을 Figure 17에서 확인할 수 있다. 본 실험에서는 이를 적절한 hyperparameter tuning을 통해 해결할 수 있었다.

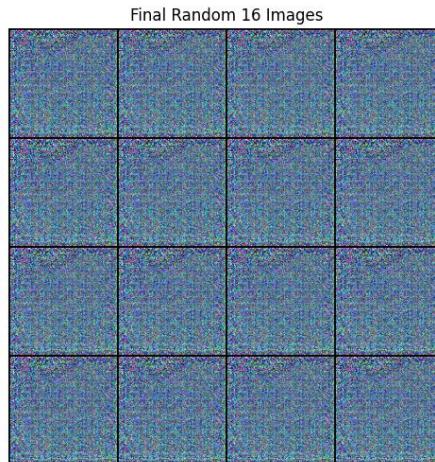


Figure 17. Image Generation with Vanishing Gradient of DCGAN

B. MNIST Case Study

Dataset Pre-processing. MNIST 데이터셋은 클래스별 약 6000장의 균형 잡힌 데이터를 포함하고 있으나[22], 본 연구에서는 극단적인 불균형 환경을 고의적으로 조성하기 위해 3개의 case로 학습 데이터셋을 Table 3처럼 재구성하였다. Case1은 클래스 1과 2의 binary classification 문제로 구성하였으며, 클래스 2를 6장만 포함하여 0.001 수준의 불균형 비율을 형성하였다. Case2는 클래스 1을 다수 클래스로, 클래스 2부터 5까지를 소수 클래스로 설정하였고 각 소수 클래스당 6장만을 포함하는 multi-class classification 문제로 구성하였다. Case3은 10개 클래스를 모두 포함하되, 클래스 0의 개수를 약 67장으로 줄여 불균형 비율을 0.01 수준으로 조정하였다. 테스트데이터셋은 모든 경우에서 클래스당 약 1000장씩으로 고정하였다.

Table 3. MNIST Dataset Pre-processing

	Objective	Majority Class	Minority Class	Imbalance Ratio
Case1	Binary CLS	1	2	0.001
Case2	Multi-class CLS	1	2~5	0.001
Case3	Multi-class CLS	0	1~9	0.01

Data Synthesis by Oversampling Methods. Oversampling Methods로는 SMOTE, B-SMOTE, ADASYN을 선택하였고, 이를 각 case1, 2, 3의 학습데이터셋에 독립적으로 적용하여 균형 데이터셋을 구성하였다. 이후 동일한 구조의 CNN을 재학습하여 precision, recall, f1-score 중심으로 baseline 대비 성능 변화를 비교하였다. 각 기법이 적용되었을 때의 성능 변화는 Table 4에 요약하였다. MNIST case study에서 진행한 실험은 case 별로 CNN 구조와 baseline 성능지표를 통일하지 못해 case by case로 진행하였다. 통일된 기준의 부재로 다소 한계점이 존재하지만 case by case로의 baseline은 명확히 구축함으로써 각 방법 적용 전후의 성능 변화를 명확히 관찰할 수 있었다.

Table 4. MNIST Data Synthesis Evaluation Results per Oversampling Methods

Method	Case	Precision	Recall	F1-score
SMOTE	case1	0.7753 → 0.7863	-	0.7604 → 0.7735
	case2	0.6005 → 0.7785	-	0.5960 → 0.7739
	case3	0.9895 → 0.9876	-	0.9895 → 0.9876
B-SMOTE	case1	0.7846 → 0.9028	0.5838 → 0.8672	0.5063 → 0.8693
	case2	0.4727 → 0.8271	0.3484 → 0.6965	0.2529 → 0.7132
	case3	0.9603 → 0.9869	0.9563 → 0.9869	0.9546 → 0.9869
ADASYN	case1	0.8957 → 0.7670	-	0.8940 → 0.7504
	case2	0.7662 → 0.8357	-	0.7668 → 0.8367
	case3	0.9863 → 0.9880	-	0.9863 → 0.9890

각 방법을 통해 생성된 이미지는 Figure 18, 19, 20을 통해 확인할 수 있다. 세 방법 모두 복제된 이미지를 들여다보면, 기존 숫자 이미지의 경계선 부근에서 blurry한 윤곽이 추가되는 모습을 볼 수 있다. 이는 특정 숫자가 가지는 명확한 직선, 곡선 등의 특징을 담아내지는 못하지만 기본적으로 극소수 불균형 문제를 해결함으로써 대부분의 성능지표가 상승한다는 것을 알 수 있었다. 생성된 이미지의 품질을 고려하면, 이미지 레벨에서의 interpolation보다는 feature space에서의 interpolation이 특정 숫자가 가지는 명확한 특징을 더 잘 표현할 수 있을 것으로 기대된다.

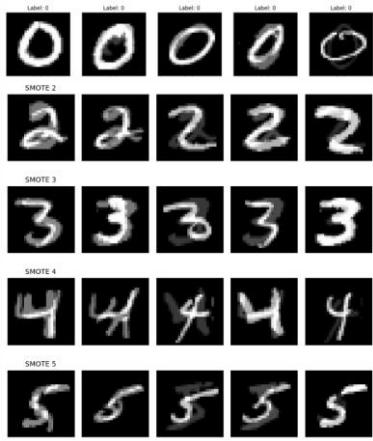


Figure 18. Synthesized MNIST Images by SMOTE

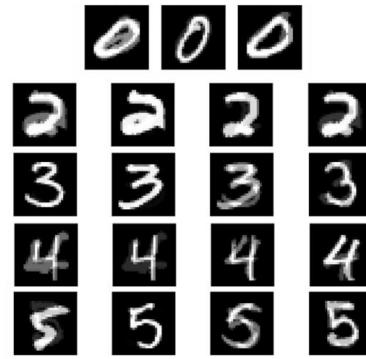


Figure 19. Synthesized MNIST Images by B-SMOTE

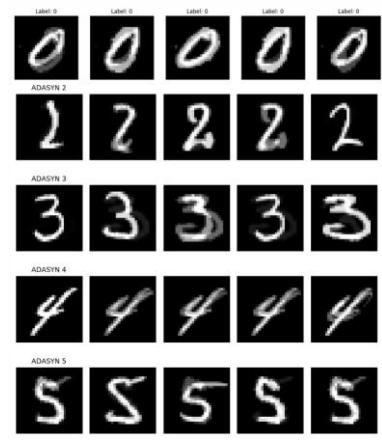


Figure 20. Synthesized MNIST Images by ADASYN

MNIST Data Synthesis by GAN and CGAN. MNIST 불균형 데이터셋의 case1과 case2의 경우, MNIST 학습데이터셋의 불균형 비율이 0.001로 매우 낮았다. 이 비율을 완화하여 학습데이터 양을 늘릴 경우 baseline 성능이 과도하게 높아지는 경우도 있었기 때문이다(case3에서도 0.01 비율로 이미 포화상태에 달하는 분류성능을 보인다). 이렇게 극단적인 학습데이터셋 구성은 batch size 설정 등 안정적인 GAN의 학습환경 조성이 어려웠고, 실제 클래스별 6장의 학습이미지로 GAN을 학습한 결과, 학습이 이루어지지 못하고 noise 이미지만을 내놓는 현상이 나타났다. 이러한 이유로 MNIST case study에서는 GAN을 직접적으로 활용하기보다는, case2에 국한하여 CGAN을 통해 data synthesis 과정을 진행하였다.

불균형 데이터셋 case2에 대해 CGAN을 적용하였기 때문에 총 학습데이터셋은 클래스별 6장, 총 24장으로 구성되었다. 200 epoch가량 학습을 진행하였음에도 Figure 21과 같이 저품질의 이미지를 생성하는 것을 확인할 수 있었다. 이미지의 품질은 매우 좋지 않았지만 GAN을 클래스별로 적용하였을 때 학습이 전혀 이루어지지 않았던 것 대비 학습 자체는 어느 정도 진행되었음을 알 수 있었다. GAN과 CGAN 모두 MNIST 데이터셋에서 성공적인 data synthesis 과정이 이루어졌다고 보기 어려웠기 때문에 별도의 CNN 분류 성능 evaluation 과정을 진행하지 않았다.

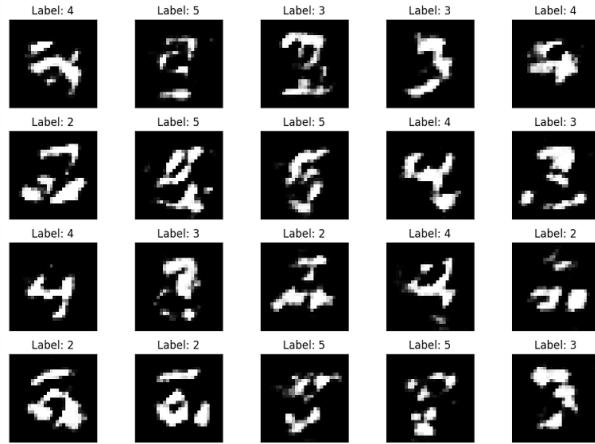


Figure 21. Synthesized MNIST Images by CGAN

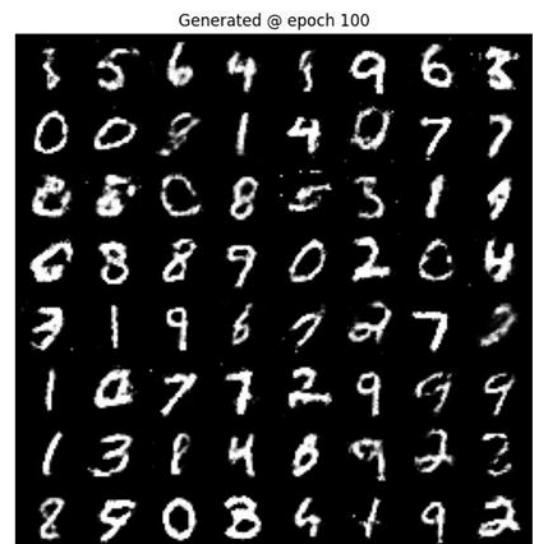


Figure 22. Generated MNIST Images by GAN with Sufficient TrainSet ($k=1$, $lr=1e-4$)

이처럼 MNIST 데이터셋에서 GAN과 CGAN 학습이 잘 이루어지지 않았던 핵심적인 이유는 학습데이터셋이 너무 극소수에 불과했다는 것으로 분석했다. 이를 확인하기 위해 불균형 형성 전 학습데이터셋으로 GAN 학습을 재진행하였고, 좋은 품질의 숫자 이미지 데이터들이 생성되는 것을 알 수 있었다. 합성된 이미지는 Figure 22를 통해 확인할 수 있다. 이 같은 결과를 AM 데이터셋에서 GAN을 적용할 때 반영하여, 적어도 학습이 진행될 수 있는 클래스별 학습데이터 수를 확보하기 위해 학습데이터셋 구성 시 추가적인 data augmentation을 진행하였다.

C. More about Hyperparameter Sensitivity of GAN with MNIST Dataset

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k , is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

```

for number of training iterations do
    for  $k$  steps do
        • Sample minibatch of  $m$  noise samples  $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$  from noise prior  $p_g(\mathbf{z})$ .
        • Sample minibatch of  $m$  examples  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$  from data generating distribution  $p_{\text{data}}(\mathbf{x})$ .
        • Update the discriminator by ascending its stochastic gradient:
            
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)})))].$$

    end for
    • Sample minibatch of  $m$  noise samples  $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$  from noise prior  $p_g(\mathbf{z})$ .
    • Update the generator by descending its stochastic gradient:
        
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(\mathbf{z}^{(i)}))).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

```

Figure 23. GAN Training Algorithm

Hyperparameter Sensitivity. Figure 23은 GAN 논문[12]에서 설명하고 있는 training 알고리즘을 표현한 자료이다. 알고리즘에는 D를 학습시키는 학습빈도에 대해 learning step k 로 hyperparameter 조정이 가능하도록 구현하였다. D와 G의 본질적인 learning rate과 더불어 이 같은 GAN의 hyperparameter는 training과 관련하여 도메인마다 적절한 값이 존재하며, 설정에 따라 학습편차가 발생한다. 우리는 이 부분에 주목하여 D와 G의 학습빈도 비율이 1:1인 MNIST 데이터셋 GAN 학습상황에서 hyperparameter tuning 실험을 통해 GAN의 hyperparameter sensitivity를 현상적으로 분석하고자 하였다. D와 G의 학습빈도 1:1, learning rate 1e-4, 100 epoch 학습을 진행하였을 때 생성하는 이미지는 Figure 22를 통해 확인할 수 있으며, D와 G의 학습 loss 그래프, G로 향하는 학습기울기 신호 등의 training details는 Figure 24, 25, 26를 통해 확인할 수 있다.

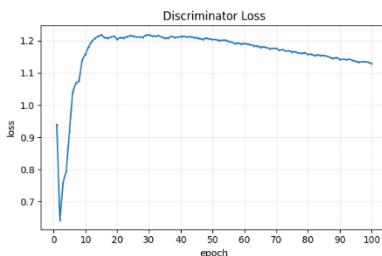


Figure 24. Discriminator Loss Graph
($k=1$, $lr=1e-4$)

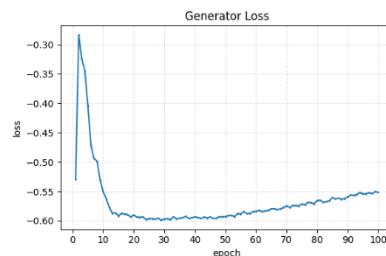


Figure 25. Generator Loss Graph ($k=1$,
 $lr=1e-4$)

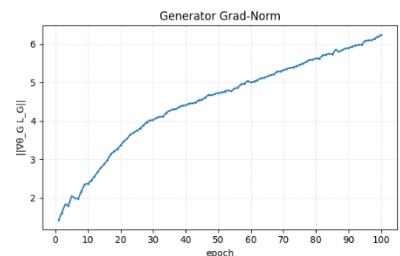


Figure 26. Gradient Norm during Generator Training ($k=1$, $lr=1e-4$)

Vanishing Gradient. 가장 먼저 D의 학습력이 강화될수록 나타나는 현상에 대해 분석하고자 D의 learning rate을 기존 및 G 대비 2배인 2e-4로 형성하고, D의 learning step k를 2로 조정하여 실험을 진행하였다. 기존에 10 epoch 정도에서 G의 학습효과가 서서히 나타났던 것에 비해 hyperparameter tuning 이후에는 20 epoch 부근에서 G의 학습효과가 서서히 나타나는 것을 알 수 있었다. 또한 D의 학습력이 강화된 만큼 G의 학습력이 다소 약화되어 이미지 생성 품질의 개선 속도가 기존 대비 다소 느려졌음을 epoch 별 생성이미지의 품질을 통해 확인할 수 있었다. 100 epoch 학습 후 생성한 이미지는 Figure 27에서 확인할 수 있고, training details는 Figure 28, 29, 30에서 확인할 수 있다.



Figure 27. Generated MNIST Images by GAN with Sufficient TrainSet ($k=2$, $D_lr=2e-4$)

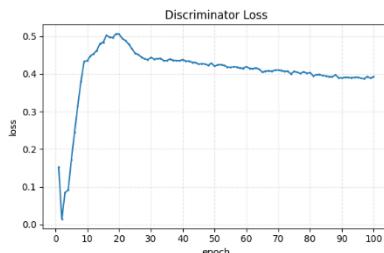


Figure 28. Discriminator Loss Graph ($k=2$, $D_lr=2e-4$)

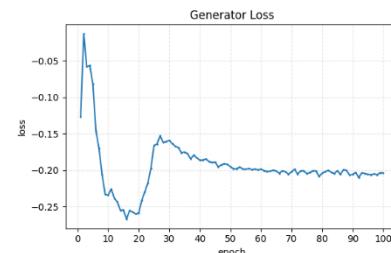


Figure 29. Generator Loss Graph ($k=2$, $D_lr=2e-4$)

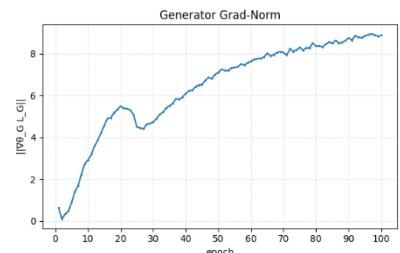


Figure 30. Gradient Norm during Generator Training ($k=2$, $D_lr=2e-4$)

이와 같이 D의 학습력을 강화할수록 G의 학습력이 더 약화되고 그만큼 이미지의 품질 개선 속도도 같이 느려짐을 확인할 수 있었다. 그러나 D의 learning step k가 4가 될 때부터는 지나친 D의 학습력으로 G에게 유효한 학습 기울기 신호를 전달하지 못해 vanishing gradient 현상이 발생함을 확인할 수 있었다. 이는 Figure 31에서 나타내는 GAN의 전형적인 loss function이 가지는 한계점으로, D의 학습력이 강화된 구간(①)에서 G의 학습기울기 신호가 강하게 전달되어야 하는데, loss function이 saturating 환경으로 구현되어 D의 학습이 더 진행될수록 G의 학습 기울기 신호가 점차 약화되는 문제로 인해 발생한다. 또 반대로 G의 학습력이 강화된 구간(②)에서는 G의 학습기울기 신호가 불필요하게 강하게 전달되어, D와 G의 균형 있는 학습이 불가능해진다. 이에 따라 학습초기부터 G로 유효한 학습기울기를 형성하지 못하는 것을 training details에 관한 Figure 33, 34, 35를 통해 확인할 수 있고, 생성되는 이미지가 noise에서 개선되지 않음을 Figure 32를 통해 확인할 수 있다.

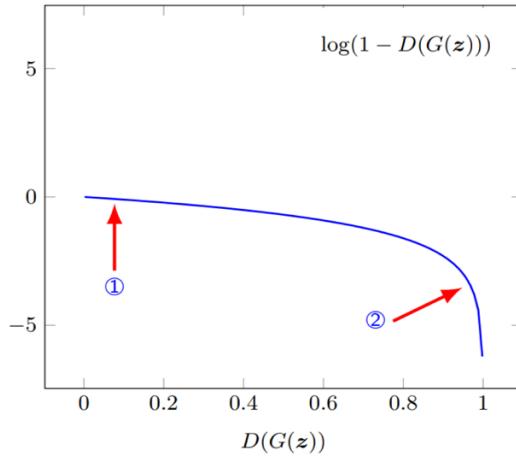


Figure 31. Saturating Loss Function of GAN

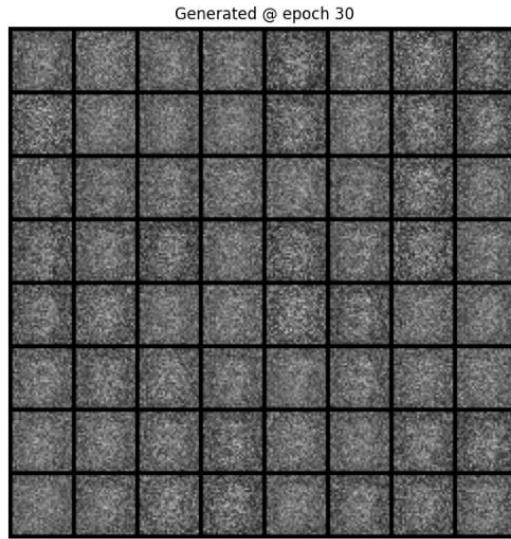


Figure 32. Generated MNIST Images by GAN with Sufficient TrainSet ($k=4$, $D_{lr}=2e-4$)

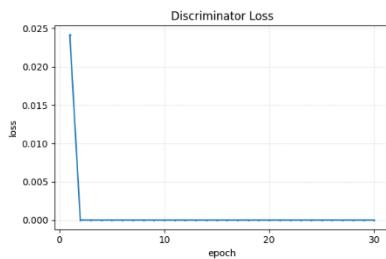


Figure 33. Discriminator Loss Graph ($k=4$, $D_{lr}=2e-4$)

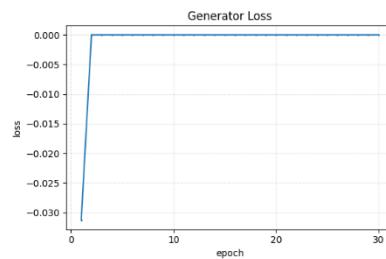


Figure 34. Generator Loss Graph ($k=4$, $D_{lr}=2e-4$)

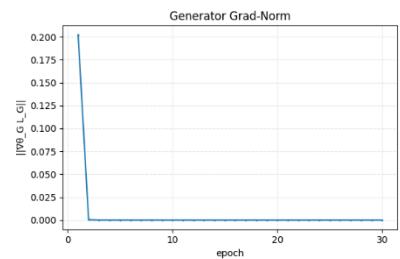


Figure 35. Gradient Norm during Generator Training ($k=4$, $D_{lr}=2e-4$)

Non-Saturating Loss Function. 앞서 마주한 vanishing gradient 문제를 해결하기 위해 GAN의 loss function 설계를 practical 측면에서 $D(G(z))$ 관련 \log 값에 대해 Figure 36처럼 non-saturating으로 전환하여 기존 Generator 입장에서 minimization 문제였던 부분을 maximization optimization 방식으로 진행할 수 있다. Loss function을 non-saturating 형태로 재설계하고 hyperparameter 설정을 $k=1$, $lr=1e-4$ 로 초기화하였을 때 나타나는 생성이미지의 품질과 training details는 Figure 37, 38, 39, 40을 통해 확인할 수 있다.

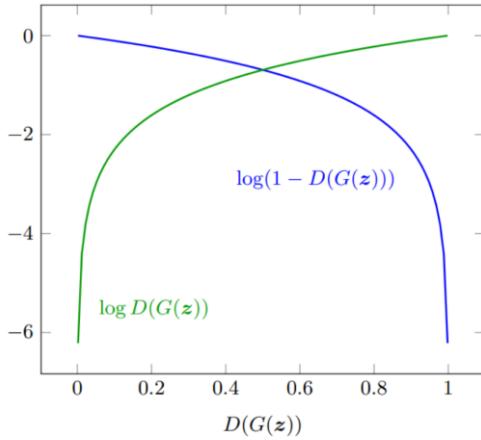


Figure 36. Non-saturating Loss Function (Green) of GAN



Figure 37. Generated MNIST Images by GAN with Non-saturating Loss ($k=1$, $lr=1e-4$)

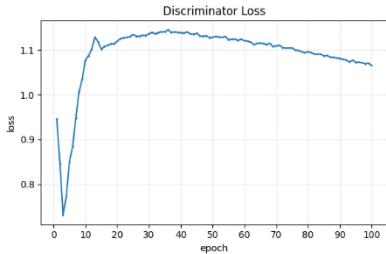


Figure 38. Discriminator Loss Graph
(non-saturating loss, $k=1$, $lr=1e-4$)

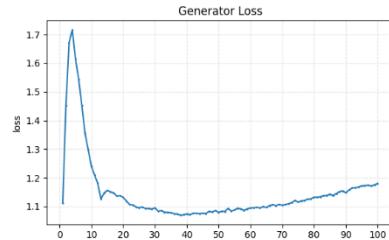


Figure 39. Generator Loss Graph
(non-saturating loss, $k=1$, $lr=1e-4$)

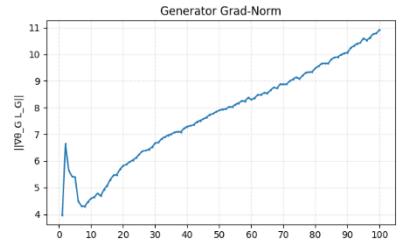


Figure 40. Gradient Norm during Generator Training (non-saturating loss, $k=1$, $lr=1e-4$)

Mode Collapse. Non-saturating function 설정 이후에도 hyperparameter sensitivity를 현상적으로 확인하기 위해 앞선 방식과 동일하게 D의 learning step k 와 learning rate을 조정하여 학습을 진행하였다. 해당 환경에서 생성된 이미지는 Figure 41과 같으며, 숫자 1만을 생성하고 있음을 확인하였다. 이는 GAN의 또 다른 근본적인 한계점으로 mode collapse 현상이며, 학습이 진행됨에 따라 G는 D를 속이기 가장 쉬운 이미지만을 생성하는데 MNIST 데이터셋에서는 많은 숫자들이 포함하고 있는 직선 형태의 가장 기본적인 특징을 가진 숫자 1만을 생성하게 되는 것으로 분석하였다. Figure 42, 43, 44의 training details를 보면 D가 G가 만들어낸 숫자1에 대한 판별력이 강하지 않아 숫자1 이미지만을 G가 생성하는 것으로 분석할 수 있었다.

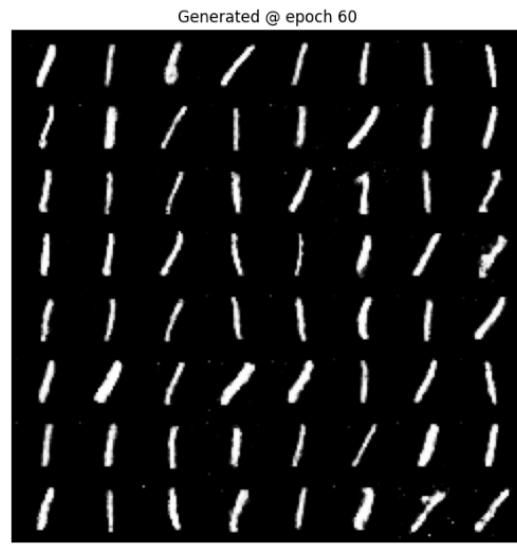


Figure 41. Generated MNIST Images by GAN with Non-saturating Loss ($k=4$, $lr_D=2e-4$)

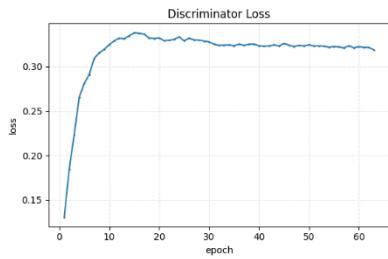


Figure 42. Discriminator Loss Graph
(non-saturating loss, $k=4$, $lr_D=2e-4$)

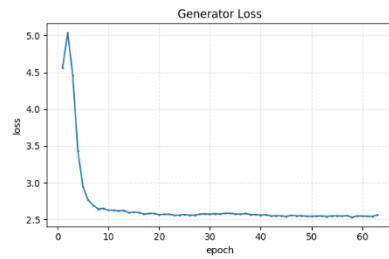


Figure 43. Generator Loss Graph
(non-saturating loss, $k=4$, $lr_D=2e-4$)

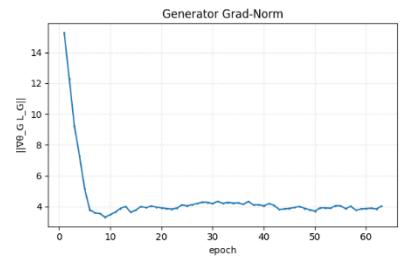


Figure 44. Gradient Norm during Generator Training (non-saturating loss, $k=4$, $lr_D=2e-4$)