

Deep Convolution Neural Networks 이용하여 결함 검출을 위한 결함이 있는 철도선로표면 디지털영상 재 생성

Regeneration of a defective Railroad Surface for defect detection with Deep Convolution Neural Networks

김 현 호 한 석 민^{1*}
Hyeonho Kim Seokmin Han

요 약

본 연구는 철도표면상에 발생하는 노후 현상 중 하나인 결함 검출을 위해 학습데이터를 생성함으로써 결함 검출 모델에서 더 높은 점수를 얻기 위해 진행되었다. 철도표면에서 결함은 선로결속장치 및 선로와 차량의 마찰 등 다양한 원인에 의해 발생하고 선로 파손(14) 등의 사고를 유발할 수 있기 때문에 결함에 대한 철도 유지관리가 필요 하다. 그래서 철도 유지관리의 자동화 및 비용절감을 위해 철도 표면 영상에 영상처리 또는 기계학습을 활용한 결함 검출 및 검사에 대한 다양한 연구가 진행되고 있다. 일반적으로 영상 처리 분석기법 및 기계학습 기술의 성능은 데이터의 수량과 품질에 의존한다. 그렇기 때문에 일부 연구는 일반적인 다양한 철도표면영상의 데이터베이스를 확보하기위해 등간격으로 선로표면을 촬영하는 장치 또는 탑재된 차량이 필요로 하였다(15, 16). 본연구는 이러한 기계적인 영상획득 장치의 운용비용을 감소시키고 보완하기 위해 대표적인 영상생성관련 딥러닝 모델인 생성적 적대적 네트워크(1)의 기본 구성에서 여러 관련연구에서 제시된 방법을 응용, 결함이 있는 철도 표면 재생성모델을 구성하여, 전용 데이터베이스가 구축되지 않은 철도 표면 영상에 대해서도 결함 검출을 진행할 수 있도록 하였다. 구성된 모델은 상이한 철도 표면 텍스처들을 반영한 철도 표면 생성을 학습하고 여러 임의의 결함의 위치에 대한 Ground-Truth들을 만족하는 다양한 결함을 재 생성하도록 설계하였다. 재생성된 철도 표면의 영상들을 결함 검출 딥러닝 모델(2)에 학습데이터로 사용한다. 재생성모델의 유효성을 검증하기 위해 철도표면데이터를 3가지의 하위집합으로 군집화 하여 하나의 집합세트를 원본 영상으로 정의하고, 다른 두개의 나머지 하위집합들의 몇가지의 선로표면영상을 텍스처 영상으로 사용하여 새로운 철도 표면 영상을 생성한다. 그리고 결함 검출 모델에서 학습데이터로 생성된 새로운 철도 표면 영상을 사용하였을 때와, 생성된 철도 표면 영상이 없는 원본 영상을 사용하였을 때를 나누어 검증한다. 앞서 분류했던 하위집합들 중에서 원본영상으로 사용된 집합세트를 제외한 두 개의 하위집합들은 각각의 환경에서 학습된 결함 검출 모델에서 검증하여 출력된 픽셀단위 분류지도 영상을 얻는다. 이 픽셀단위 분류지도영상들과 실제 결함의 위치에 대한 원본결함 지도(Ground-Truth)들의 IoU(Intersection over Union) 및 F1-score로 평가하여 성능을 계산하였다. 결과적으로 두개의 하위집합의 텍스처 영상을 이용한 재생성된 학습데이터를 학습한 결함 검출모델의 점수는 원본 영상만을 학습하였을 때의 점수보다 약 IoU 및 F1-score 가 10~15% 증가하였다. 이는 전용 학습 데이터가 구축되지 않은 철도표면 영상에 대해서도 기존 데이터를 이용하여 결함 검출이 상당히 가능함을 증명하는 것이다.

☞ 주제어 : 철도표면, 생성적 적대적 네트워크, 영상 생성, 조건부 생성모델, 검출 모델

ABSTRACT

This study was carried out to generate various images of railroad surfaces with random defects as training data to be better at the detection of defects. Defects on the surface of railroads are caused by various factors such as friction between track binding devices and adjacent tracks and can cause accidents such as broken rails (14), so railroad maintenance for defects is necessary. Therefore, various researches on defect detection and inspection using image processing or machine learning on railway surface images have been conducted to automate railroad inspection and to reduce railroad maintenance costs. In general, the performance of the image processing analysis method and machine learning technology is affected by the quantity and quality of data. For this reason, some researches require specific devices or vehicles to acquire images of the track surface at regular intervals to obtain a

¹ Department of Computer Science & Information Engineering,
Korea National University of Transportation, Uiwang, 16106,
Korea.

* Corresponding author: (seokmin.han@ut.ac.kr)

[Received 29 April 2020, Reviewed 17 June 2020(R2 27 July 2020), Accepted 14 September 2020]

database of various railway surface images [15, 16]. On the contrary, in this study, in order to reduce and improve the operating cost of image acquisition, we constructed the 'Defective Railroad Surface Regeneration Model' by applying the methods presented in the related studies of the Generative Adversarial Network (GAN) (1). Thus, we aimed to detect defects on railroad surface even without a dedicated database. This constructed model is designed to learn to generate the railroad surface combining the different railroad surface textures and the original surface, considering the ground truth of the railroad defects. The generated images of the railroad surface were used as training data in defect detection network (2), which is based on Fully Convolutional Network (FCN) (3). To validate its performance, we clustered and divided the railroad data into three subsets, one subset as original railroad texture images and the remaining two subsets as another railroad surface texture images. In the first experiment, we used only original texture images for training sets in the defect detection model. And in the second experiment, we trained the generated images that were generated by combining the original images with a few railroad textures of the other images. Each defect detection model was evaluated in terms of 'intersection of union(IoU)' and F1-score measures with ground truths. As a result, the scores increased by about 10~15% when the generated images were used, compared to the case that only the original images were used. This proves that it is possible to detect defects by using the existing data and a few different texture images, even for the railroad surface images in which dedicated training database is not constructed.

☞ keyword : Railroad surface, Generative Adversarial Network, Image Representation, Conditional generation model, Detection Model

1. 서 론

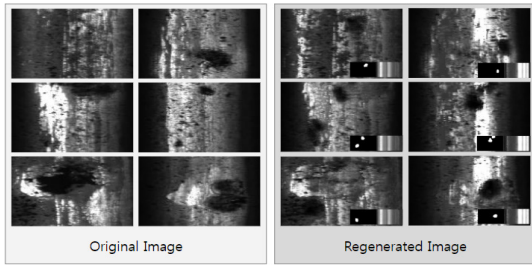
철도는 전철 및 지하철이 운행하는 선로로서 철도안전관리자는 안전사고예방을 위해 철도의 노화 및 피로수명을 고려하여 교체 및 정비를 시행하여야 한다. 철도의 결함은 인접선로 및 결속품들의 마찰 등 다양한 원인에 의해 발생한다. 철도의 결함을 관리하기 위해 국제철도연맹(International Union of Railways: UIC)은 철도의 결함의 발생 위치 및 특성에 따라 철도의 결함을 표준화하고 관리 방법 등을 제시하였으며[17] 한국의 경우 철도공단 사규의 내부지침에 따라 이를 관리하고 있다. 다양한 철도의 결함에서 철도표면상의 결함은 초기에는 육안으로 식별이 어려우나 일반적으로 시간이 지남에 따라 크기가 확장하여 철도의 손상을 유발한다[18]. 그렇기 때문에 UIC는 결함의 크기 및 깊이별로 결함이 발생한 철도선로를 교체 및 정비하고 있다[19]. 이러한 이유로, 선로의 결함검사가 주기마다 필요하고 결함의 대한 크기의 측정이 시각적으로 정확해야 한다.

본 연구는 결함 검출 모델의 검출능력 향상을 위해 결함이 존재하는 선로표면영상을 재생성하여 전용 데이터베이스를 구축하였다[2]. 이를 위해 Deep Convolution Neural Network (DCNN)기반의 생성모델과 탐지모델을 설계하고 결함을 철도표면 상에 임의로 생성하여 실제 결함을 탐지함으로써 더 높은 신뢰도를 얻도록 하였다. 생성 모델은 철도표면상에 결함을 표현하고 다양한 텍스처를 반영하기 위해 결함이 있는 일부 철로표면의 영상에서 또 다른 형태의 철도표면 텍스처와 융합된 형태의

철도표면영상을 재 생성한다. 결함은 조건부 정보로 결함의 대한 위치를 나타내는 임의의 이진화 영상을 사용하여 결과적으로 다양한 철도표면 영상을 생성하여 전용 데이터베이스를 구축할 수 있었다.

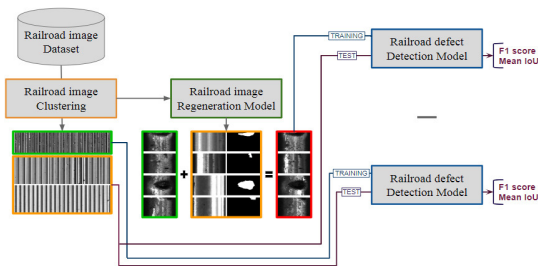
Generation Adversarial Network(GAN) [1]을 기반으로 다양한 목적에 따라 개선된 연구들이 진행되었으며(e.g., [4, 5, 6, 8, 9]), 특히 Deep Convolutional Generative Adversarial Network(DCGAN) [4]는 보다 구체적이고 깊은 회선처리 기반의 네트워크 구조를 제안하였다. 또한 Conditional GAN (CGAN) [5]에서 임의의 조건부 정보를 학습에 사용함으로써 조건에 부합하는 원하는 영상을 생성하였다. 본 연구는 DCGAN을 기반으로 전체적인 생성 모델을 설계하였고 결함의 위치를 표현하는 Ground-Truths를 학습에 사용하여 임의의 위치에 결함이 존재하는 철도표면영상을 생성하였다. 또한 Pix2Pix [6]에 근거하여 생성 모델이 원본 영상과 일부 텍스처영상을 융합하도록 하였다(그림 1).

본 연구의 개략적인 개념은 그림 2와 같으며, 전용 데이터 베이스에 포함되지 않은 철도 표면의 결함을 찾아내는 것이다. 이를 위하여, 원본 철도표면영상과 데이터 베이스로 포함하지 않은 철도표면의 일부 영상을 융합하여 새로운 결함이 존재하는 철도표면영상을 재 생성한다. 재 생성된 영상들로 새로운 데이터베이스를 구축하여 전용 실제 영상 데이터베이스가 구축되지 않은 철로표면에 대해서도 결함 검출 모델이 결함을 검출할 수 있도록 한다.



(그림 1) 임의의 철도 표면과 결함의 위치정보영상에 의한 철도 표면 재생성

(Figure 1) Image Regeneration by custom railroad surfaces and Ground-Truths



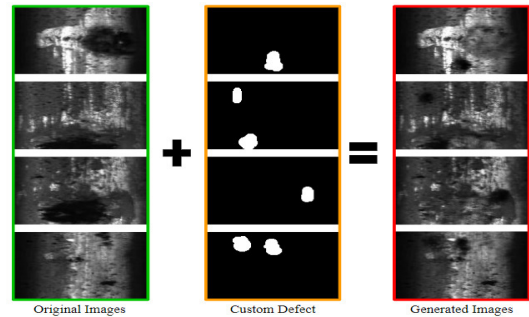
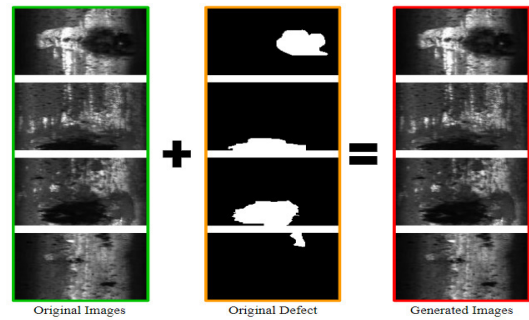
(그림 2) 전체 연구의 개념도. 원본 철도 표면 영상을 이용한 철도 표면 결함 검출 및 재 생성된 철도 표면 영상을 이용한 철도 표면 결함 검출

(Figure 2) Conceptual diagram of the entire research. Railroad surface defect detection comparison using regeneration railroad surface image and versus original railroad surface image

2. 철도표면 영상 재 생성모델 개요

2.1 텍스처를 반영하는 철도표면 재 생성

새로운 철도 표면 영상 생성은 텍스처영상을 이용하여 사용하고 원본영상과 텍스처영상 간의 혼합된 형태로 생성하는 개념이다. 또한 원본 결함 지도를 입력하여 생성한 결과와 임의의 결함 지도를 입력하여 생성한 결과로 나눌 수 있다. 임의의 결함을 입력하는 것은 제한이 없는 다양한 철도 표면 영상을 생성할 수 있음을 의미한다. 본 연구에서 생성한 새로운 철도 표면 영상은 다음 그림 3과 같으며 녹색 영역은 원본 철도표면영상, 적색 영역은 생성된 철도 표면 영상이다.

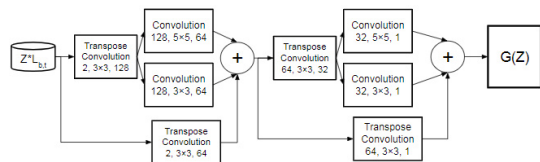


(그림 3) 원본 철도 표면 영상과 또 다른 철도 표면 영상을 혼합하고 기존 또는 임의의 결함 지도를 입력한 결과

(Figure 3) The result of blending the original railroad surface image and another railroad surface image and inputting an existing or custom defect

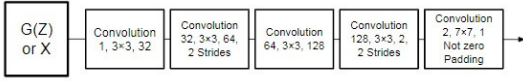
2.2. 재 생성 모델의 구조와 주요 개념

본 연구에서 생성기는 Aggregated Residual Network (ResNeXt) [7]에서 제안한 구조에서 변형하여 구성하였다(그림 4). 분류기는 분류를 위한 회선 신경망(CNN; Convolutional Neural Network)의 형태로 구성하였다(그림 5).



(그림 4) Aggregated Residual Network를 변형하여 구성한 생성기의 구조

(Figure 4) Generator architecture constructed by transformed Aggregated Residual Network



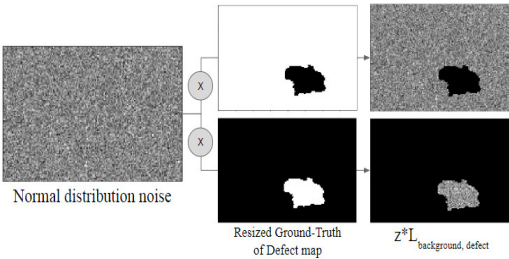
(그림 5) 단순한 분류를 위한 회선 신경망의 형태로 구성된 분류기의 구조

(Figure 5) Discriminator architecture in the form of Convolution Neural Network for simple classification

2.2.1. 생성기의 세부사항

생성기는 분류기와 경쟁하며 실제영상과 가까운 가짜 영상을 생성하며 본 연구에서 생성기를 구성할 때의 주요 개념은 다음과 같으며 세부 사항은 표 1과 같다.

Conditional information 입력된 Ground-Truths, 의 크기를 사용될 Noise, z와 같은 크기인 [w/4, h/4]로 줄이고 Noise, z를 의 각 차원 간에 곱하여 조건부 정보를 정의한다(그림 6).



(그림 6) 조건부 정보의 정의

(Figure 6) Definition of conditional information

Transpose Convolution and Batch Normalization, Relu DCGAN은 생성기에서 Transpose Convolution으로 Upscaling을 하였으며 레이어간 활성화 함수로 Rectified Linear Unit(Equation. 1)의 사용과 안정적인 학습을 위해 Batch Normalization layers의 사용을 제안하였다.

Rectified Linear Unit:

$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \quad (1)$$

Min Max Scaling after Tanh 생성기의 종단에 Hyperbolic tangent function(Equation. 2)과 Min-Max Normalization(Equation. 3)을 사용하며 생성기의 출력을 최소-최대값(0, 1 or255)으로 표현하였으며, 이는 직접적

으로 텍스트영상과 원본 영상간의 유사도를 연산 하여 비교하기 위함이다[Section 2.3.1].

Hyperbolic Tangent:

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (2)$$

Min-Max Normalization:

$$f(x) = (newMax - newMin) \frac{x - \min(x)}{\max(x) - \min(x)} + newMin \quad (3)$$

Aggregated Convolution ResNeXt에서 제안된 Residual Layer의 Filter의 크기를 3x3으로 사용하고 Convolution Layer에서 3x3 filter와 5x5 filter로 두개의 filters를 병렬적으로 사용하였다.

Expanded Bias 바이어스를 레이어의 출력크기만큼 확장하여 출력의 전체 영역에 대해 개별적인 Bias를 더함으로써 일괄 적인 밝기가 주어지는 현상을 방지한다.

(표 1) 생성기의 세부사항

(Table 1) The detail of generator

Generator			
#	Layers	# of Weight	Output
1	X*Resize(L _{background, target})	-	[n, H/4, W/4, 2]
2	Transpose Convolution, Batch Normalization, Relu	2×3×3×128 + H/2×W/2×128 + 2	[n, H/2, W/2, 128]
3	Transpose Convolution	2×3×3×64	[n, H/2, W/2, 64]
4	Σ [Convolution] [Convolution] / Batch Normalization, Relu	Σ [128×3×3×64] [128×3×3×64] + H/2×W/2×64 + 2	[n, H/2, W/2, 64]
5	Σ [(3), (4)]	-	[n, H/2, W/2, 64]
6	Transpose Convolution, Batch Normalization, Relu	64×3×3×32 + H/2×W/2×32 + 2	[n, H, W, 32]
7	Transpose Convolution	64×3×3×1	[n, H, W, 1]
8	Σ [Convolution] [Convolution] / Relu	Σ [32×3×3×1] [32×3×3×1] + H×W×1 + 2	[n, H, W, 1]
9	Σ [(7), (8)], Tanh, Min-Max Scaling	-	[n, H, W, 1]

2.2.2. 분류기의 세부사항

분류기는 실제영상과 생성된 영상이 진짜인지 가짜인지 분류하며 생성기의 목표는 이러한 분류기를 속이는 영상을 생성하는 것이다. 본 연구에서 구성한 분류기의 주요 개념은 다음과 같으며 세부사항은 Table 2와 같다.

Convolution with strides and Batch Normalization, Leaky-Relu DCGAN은 Pooling layer를 Strides 2이상의 Convolution layer로 대체하고 분류기의 각 레이어의 활성화 함수로 Leaky-Relu(Equation. 4)의 사용을 제안하였다.

Leaky Rectified Linear Unit:

$$f(x) = \begin{cases} ax & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}, a = 0.2 \quad (4)$$

Sigmoid and Patches Discriminate Pix2Pix [6]에서 분별기의 종단에 $N \times N$ 의 일정한 크기의 영역별로 참 또는 거짓을 분류할 때 영역 공간의 형태의 참 또는 거짓을 분류하는 것으로 볼 수 있다고 하였다. 이에 근거하여 우리는 분류기 종단의 Convolution Filter size, N 을 7로 설정하여 영역별로 분류하고 활성화 함수로 Sigmoid Function(Equation. 5)을 사용한다.

Sigmoid Function:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

(표 2) 분류기의 세부사항

(Table 2) The detail of discriminator

Discriminator			
#	Layers	# of Weight	Output
1	Convolution, Leaky-Relu	$1 \times 3 \times 3 \times 32 + 32$	[n, H, W, 32]
2	Convolution with strides 2, Batch Normalization, Leaky-Relu	$32 \times 3 \times 3 \times 64 + 64 + 2$	[n, H/2, W/2, 64]
3	Convolution, Batch Normalization, Leaky-Relu	$64 \times 3 \times 3 \times 32 + 32 + 2$	[n, H/2, W/2, 32]
4	Convolution with strides 2, Batch Normalization, Leaky-Relu	$32 \times 3 \times 3 \times 2 + 2 + 2$	[n, H/4, W/4, 2]
5	Convolution with not zero padding, Sigmoid	$2 \times 7 \times 7 \times 1 + 1$	[n, H/4, W/4, 1]

2.3. Training Method

2.3.1. Least Squared Loss and L1-Norm, One-side Smooth Labels

Ref[8, 9]를 참조하여 Least Squared Loss function을 기반으로 학습하며, 여기에 One-side smoothing을 적용하면 다음과 같다.

Least squared Loss[9] + One-side smoothing[8]:

$$\begin{aligned} \min D &= \frac{1}{2} E_{x \sim p_{\text{data}}(x)} [(D(x) - 0.9)^2] \\ &+ \frac{1}{2} E_{z \sim p_z(z)} [(G(D(z)) - 0)^2] \\ \min G &= \frac{1}{2} E_{x \sim p_z(z)} [G(D(z)) - 0.9]^2 \end{aligned} \quad (6)$$

그리고 Equation. 6에서 본연구는 분류기에서 텍스처 영상에 대해 추가로 분류하고 생성기에서 Pix2Pix에서 제시한 L1-Norm을 원본영상과 텍스처영상에 대해 추가로 계산한다 (Equation. 7). 여기서 L1-Norm은 생성기에 주어지는 방향성이라고 할 수 있다.

Our final loss function:

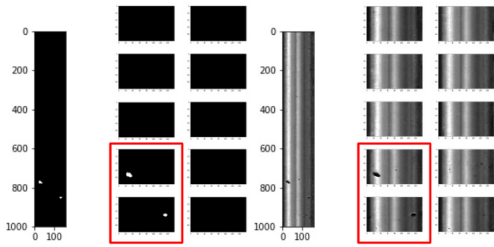
$$\begin{aligned} \min D &= \frac{1}{2} E_{z \sim p_z(z), L \sim p_{\text{data}}(L)} [(D(G(z, L)))^2] \\ &+ \frac{1}{2} E_{x \sim p_{\text{data}}(x)} [(D(x) - 0.9)^2] \\ &+ \beta E_{s \sim p_{\text{sgb}}(s), L \sim p_{\text{data}}(L)} [(D(s \cdot L_0) - 0.9)^2] \\ \min G &= \frac{1}{2} E_{z \sim p_z(z), L \sim p_{\text{data}}(L)} [(D(G(z, L)) - 0.9)^2] \\ &+ \frac{1}{2} E_{x \sim p_z(z), L \sim p_{\text{data}}(L), x \sim p_{\text{data}}(x), s \sim p_{\text{sgb}}(s)} \\ &[\alpha |G(z, L) - x| + \beta |G(z, L) - (s \cdot L_0)|] \end{aligned} \quad (7)$$

최종 손실 함수, Equation. 7에서 α, β 는 재 생성된 영상의 각각 원본영상과 텍스처영상의 반영률을 의미할 수 있으며 $\alpha = 1, \beta = 1$ 로 정의하였다.

3. Implementation

3.1. 데이터베이스

본 연구에서 사용한 데이터베이스[12]는 주행용 철도 표면을 일정구간 촬영한 8-bit Grayscale 영상이며 결함의 위치를 나타내는 Ground-Truths가 존재한다. 철도표면영상의 값의 분포를 0과 1사이로 정규화 하여 사용하며 각 철도표면영상과 Ground-Truths의 집합은 100 Pixel간 나누어 사용하였으며 결함이 존재하는 철도 표면영상을 선택하여 학습에 사용하였다 (그림 7)



(그림 7) 결함이 존재하는 철도 표면영상만을 선택하여 데이터 베이스 구축

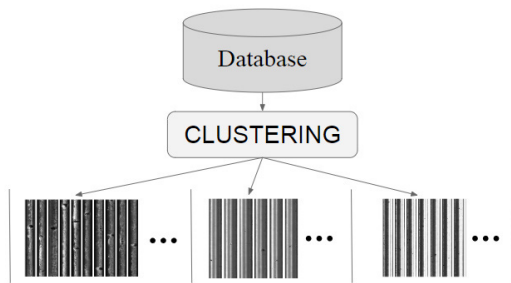
(Figure 7) A definition of database, only used defective railroad surface images

3.2. 생성 모델의 가중치 초기화 방법

DCGAN은 생성기의 가중치 초기화방법으로 $N(0,0.02)$ 의 정규분포 사용을 제안하였으며 본 연구 역시 이를 사용하였다. 또한 Batch Normalization layer에서 Zero divide 문제를 피하기 위해 variance epsilon으로 5e-5를 계산하였다.

3.3. 데이터 군집화

생성할 영상의 재료가 되는 원본영상과 텍스처영상은 기존의 데이터베이스에서 영상 군집화 방법[13]에서 제안한 방법으로 n개의 군집으로 구분하여 사용하며 본 연구는 가장 많은 개수를 갖고 있는 집합세트를 원본영상으로 사용하며 나머지 서브세트들의 몇 가지 영상을 텍스처영상으로 사용하여 새로운 철도표면 영상을 생성하였다(그림 8).



(그림 8) 데이터 군집화

(Figure 8) Data clustering

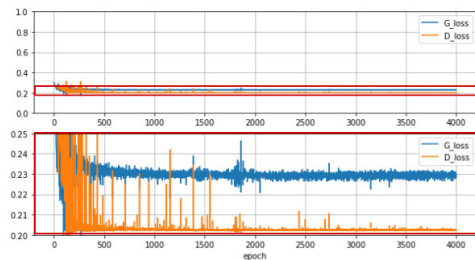
4. Experiment

4.1. RMSProb에 의한 최적화

LSGAN[9]는 손실 값 최적화 방법으로 각각 Adam[10]과 RMSProp[11] 두 가지를 실험 및 비교하였다. 해당 연구의 실험결과에서 학습률(Learning Rate)을 낮은 값인 0.0002를 사용하고 Adam을 사용한다면 관성에 관여하는 초 매개변수, $\beta_1 = 0.5$ 로 설정하거나 RMSProp을 선택하는 것이 모델의 안정적인 학습을 보여준다고 하였다. 본 연구는 이에 따라 RMSProp 최적화방법을 사용하며 학습률로 0.0002를 사용한다.

4.2. 실험의 환경과 관측결과

생성할 수 있는 임의의 영상은 임의의 텍스처를 적용하거나 제외할 수 있다. 실험은 GPU로는 GTX 1080(8GB)을 사용하고 결함이 존재하는 53개의 원본영상을 사용하며 Batch-size는 16으로 실험하였다. 실험은 4000번의 시행(시행간 약 0.9초), 즉 약 1시간동안 학습을 진행하였다. 그림 9는 생성기와 분류기의 시행간 손실값을 보여주며 2000번의 시행 이후 손실 값의 변화량이 크지 않아 2000번의 시행에서 원하는 임의의 영상을 생성하였다.



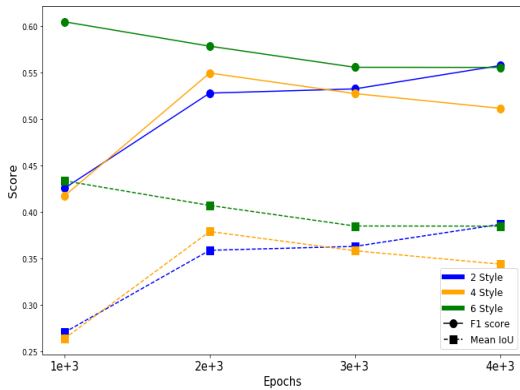
(그림 9) 생성기(파란색) 및 분류기(주황색)의 학습 시행간 손실 그래프

(Figure 9) Training loss graph of Generator (Blue) and Discriminator (Orange)

생성한 철도 표면 영상을 검증하기 위해 철도 표면의 결함을 검출하는 모델에 원본 영상을 학습하거나 생성된 철도 표면 영상을 학습하여 나머지 집합세트에 대해 결함을 검출한다. 픽셀단위 분류문제에서 모델을 평가하기 위한 점수로 결함 영역에 대해서 Mean Intersection over Union(Mean IoU)와 F1-score로 평가하였다. 이것은

데이터의 전체영역에서 결함이 차지하는 비율은 약 1% 이하로 극히 적기 때문이다.

우리는 생성모델의 학습에 사용되어 혼합될 텍스처영상의 개수와 학습 횟수 사이의 Trade-off 현상을 관찰할 수 있었다. 또한 공통적으로 재생성모델에서 과도한 횟수 또는 의도적으로 편향적인 학습(e.g. $| - | > 0.5$)을 진행하였을 때 원본영상 또는 텍스처영상으로 과적합이 이루어질 수 있음을 확인하였다. 이로 인해 우리는 임의의 텍스처영상을 총 4개를 사용하였고 학습 횟수를 손실 값의 변화량이 크지 않은 2000번(약 0.5시간)에서 임의의 철도 표면영상을 재 생성하였다. 그림 10 은 사용한 텍스처의 개수와 반복횟수마다 결함 검출 모델의 점수를 보여주며 사용한 텍스처영상의 개수 및 반복횟수가 길어질 경우 결함 검출 모델에서 점수가 하락하는 것으로 관측하였다.

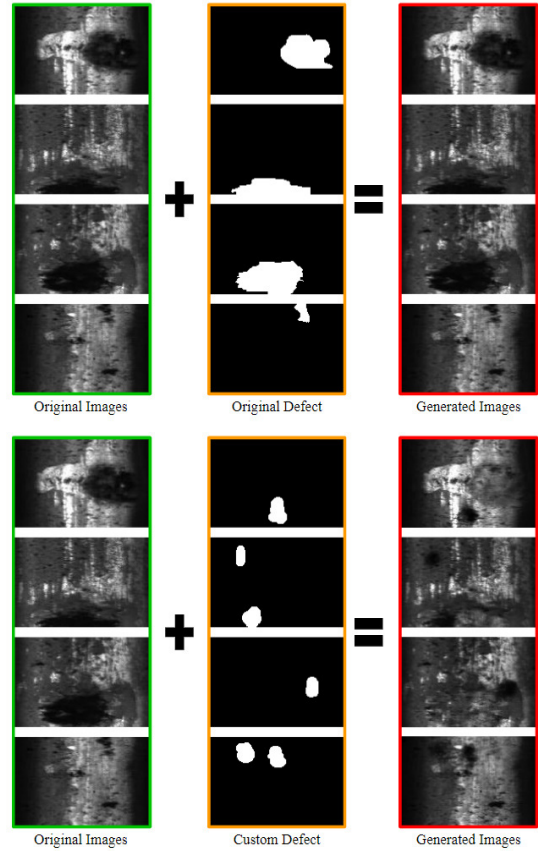


(그림 10) 생성 모델의 반복 횟수와 혼합될 텍스처영상의 개수에 따른 검출 모델의 점수

(Figure 10) The scores of the detection model according to the number of iterations of the regeneration model and the number of textures to be blended.

4.3. 원본영상으로 재 생성

Section 2.3.1의 Equation. 7에서 텍스처의 반영률이라고 볼 수 있는 λ 를 0으로 학습을 하면 곧 원본 철도표면 영상 과 동일한 철도표면 영상을 생성하게 되며 생성 영상과 원본 영상의 차이를 사람의 눈으로 구별하기 어려워지며(그림 11 상단) 생성된 영상과 기존영상간 L1-Norm 및 L2-Norm는 각각 $1.9e-5$, $3.8e-3$ 으로 매우 낮음으로 상당히 유사하다.



(그림 11) 원본 철도표면 영상에 생성하고 기존 또는 임의의 결함을 입력하여 생성한 영상

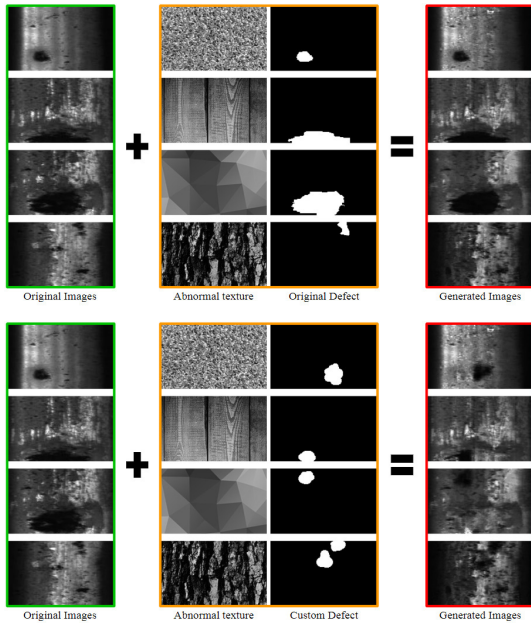
(Figure 11) Result of blending Original images with the normal rail-surface textures and inputting Original or Custom defect map

4.4. 일반적이지 않은 텍스처영상으로 재 생성

일반적인 철도영상이 아닌 텍스처영상에 대해서 생성한 결과는 다음과 같으며 텍스처의 패턴이 뚜렷할수록 더 잘 반영된 것을 보여주었다.

5. 결론 및 향후 과제

본 연구는 결함 검출에 앞서 한정된 철도 표면 데이터의 신뢰성이 높은 확장 방법을 찾고 더 높은 성능의 철도 표면상의 결함 검출을 위해 진행되었다. 본 연구의 생성 모델은 다양한 텍스처를 표현하기 위해 임의의 표면 영



(그림 12) 일반적이지 않은 Textures에 대해 원본 철도표면 영상과 혼합하고 기존 또는 임의의 결함 지도가 입력된 결과
 (Figure 12) Result of generation to Original with Original or Custom defect maps

상과 본래 영상을 불규칙적으로 결합하는 GAN기반의 생성 모델을 구성하였다. 또한 결함이라는 특정 공간을 그려낼 수 있도록 하기 위해, 조건부 정보를 사용하여 원하는 영상을 생성하는 Conditional GAN[2]이 제시한 방법을 적용하였다. 본 연구의 생성 모델은 시각적으로 변화가 있으면서도, 사람의 눈으로는 진짜인지 가짜인지 분별하기 어렵다고 생각되는 임의의 철도 표면 영상을 생성할 수 있었다. 생성된 철도 표면 영상은 기존 검출 모델의 학습에 사용될 수 있었으며 이러한 생성된 철도 표면에 대해서 결함을 검출하고 측정한 F1 Measure, Mean IoU 등의 평가점수가, 사용한 임의의 텍스처영상의 개수에 따라 소폭 상승한 것을 확인할 수 있었다. 그러나 학습 과정간 원본영상의 결합에 대한 Ground-Truth만을 사용하여서 학습이 오래 지속될수록 결함에 대한 영역이 텍스처영상으로 과적합되어 결과적으로 원하는 결함보다 작은 영역의 결함을 표현하였다. 하지만 본 연구에서 반복횟수를 적절히 제한하여 생성한 철도표면영상이 실제로 단독으로도 학습에 사용되어 긍정적인 결과를 보여주었다.

Acknowledgements

이 논문은 2020년도 한국교통대의 지원과, 2020년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2017R1C1B5077068).

참고문헌(Reference)

- [1] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.
<https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [2] Hyeonho Kim, Suchul Lee, and Seokmin Han. "Railroad Surface Defect Segmentation Using a Modified Fully Convolutional Network.", KSII Transactions on Internet and Information Systems, submitted, 2020.
- [3] Jonathan Long, Evan Shelhamer, Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.
<https://doi.org/10.1109/TPAMI.2016.2572683>
- [4] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks.". 2015.
<https://arxiv.org/abs/1511.06434>
- [5] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets.". 2014.
<https://arxiv.org/abs/1411.1784>
- [6] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
<https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.632>
- [7] Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
<https://doi.org/10.1109/CVPR.2017.634>

- [8] Salimans, Tim, et al. "Improved techniques for training gans." Advances in neural information processing systems. 2016. <https://arxiv.org/abs/1606.03498>
- [9] Mao, Xudong, et al. "Least squares generative adversarial networks." Proceedings of the IEEE International Conference on Computer Vision. 2017. <https://arxiv.org/abs/1611.04076>
- [10] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization.". 2014. <https://arxiv.org/abs/1412.6980>
- [11] Tieleman, Tijmen, and Geoffrey Hinton. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." COURSE: Neural networks for machine learning, 4, 26-31, 2012.
- [12] Jinrui Gan, Q.Y. Li , et al. "A Hierarchical Extractor-Based Visual Rail Surface Inspection System". IEEE Sensors Journal, 17(23), 7935-7944, 2017. [Online].Available: <http://icn.bjtu.edu.cn/Visint/resources/RSDDs.aspx>
- [13] Sleit, Azzam, et al. "Image Clustering using Color, Texture and Shape Features." KSII Transactions on Internet & Information Systems 5.1. 2011. <https://doi.org/10.3837/tiis.2011.01.012>
- [14] Schafer, Darwin H. "Effect of train length on railroad accidents and a quantitative analysis of factors affecting broken rails." Diss. University of Illinois at Urbana-Champaign. 2008. http://railtec.illinois.edu/wp/wp-content/uploads/pdf-archive/Final-Schafer_Thesis_2008.pdf
- [15] Choi, Woo-Yong, Jeong-Yeon Kim, and Il-Dong Yang. "Rail Surface Defect Detection System of Next-Generation High Speed Train." The Transactions of The Korean Institute of Electrical Engineers, 66.5, 870-876, 2017. <https://doi.org/10.5370/KIEE.2017.66.5.870>
- [16] Min, Yongzhi, et al. "Real time detection system for rail surface defects based on machine vision." EURASIP Journal on Image and Video Processing, 2018. <https://link.springer.com/article/10.1186/s13640-017-0241-y>
- [17] International Union of Railways (UIC). "UIC 712: Rail defects", 2002.
- [18] R Andersson, Robin. "Surface defects in rails-Potential influence of operational parameters on squat initiation", 2015. <http://publications.lib.chalmers.se/records/fulltext/216733/216733.pdf>
- [19] International Union of Railways (UIC). "UIC 725: Treatment of Rail Defects", 2007.

● 저 자 소 개 ●



김 현 호(Hyeonho Kim)

2018년 국립한국교통대학교 컴퓨터정보공학과(공학사)
 2018년~현재 한국교통대학교 일반대학원 컴퓨터정보공학과(석사과정)
 관심분야 : 인공지능, 영상처리, 머신러닝
 E-mail : rlagusgh125@ut.ac.kr



한 석 민(Seokmin Han)

2000년 서울대학교 전기공학부(공학사)
 2003년 서울대학교 대학원 전기컴퓨터공학부(공학석사)
 2008년 서울대학교 대학원 전기컴퓨터공학부(공학박사)
 2017년~현재 한국교통대학교 컴퓨터정보공학과 부교수
 관심분야 : 인공지능, 영상처리, 의료영상.
 E-mail : seokmin.han@ut.ac.kr