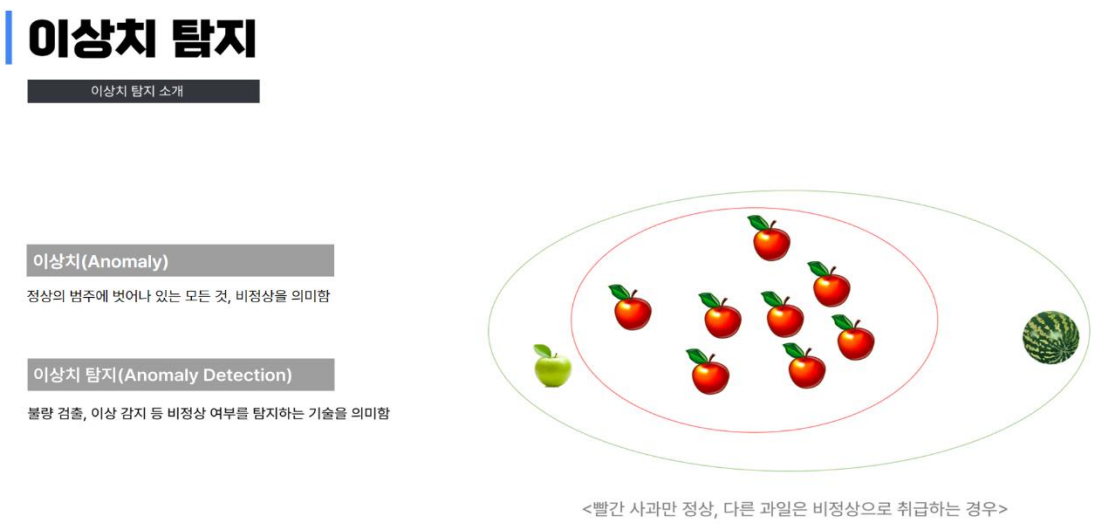
시스템보안 발표 대본

Anomaly Detection에 대한 발표를 시작하겠습니다.
PT는 이상치 탐지, GAN기반 모델, 모델 활용, 실습으로 구성하였습니다.

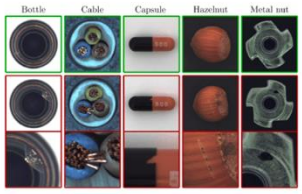


이상치란 정상의 범주에 벗어나 있는 모든 것, 즉 비정상을 의미합니다.
그리고 이상치 탐지는 불량 검출, 이상 감지 등 비정상 여부를 탐지하는 기술을 의미합니다.
예를 들어 과일이 나열된 사진에서 빨간 사과만 정상이고 나머지 과일은 모두 비정상으로 취급해 보겠습니다. 그렇다면 빨간 사과만 감싸는 경계선을 만들어서, 경계선 안쪽은 정상, 바깥쪽은 비정상으로 취급하는 방식으로 이상치 탐지를 수행할 수 있습니다.

이상치 탐지

이상치 탐지 활용 예제

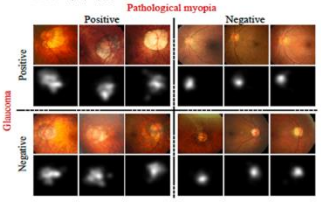
MVTec 데이터 셋



[제조] 불량 제품 탐지

제조 환경에서 불량 검출을 위한 데이터셋이다.
이미지 수는 약 5000장으로(정상 3 : 비정상 2),
15개의 카테고리를 지닌다.

LAG 데이터셋



[의료] 녹내장 구별

녹내장 검출을 위한 의료 진단 데이터셋이다.
이미지 수는 약 6000장이다. (정상 3 : 비정상 2)

UCSD 데이터셋

[보안] 이상 상황 탐지

영상 감시를 위한 CCTV 데이터 셋이다.
비디오 수는 약 100개이다. (정상 1 : 비정상 1)

이러한 이상치 탐지 기술은 제조, 의료, 보안 등 다양한 영역에서 활용되고 있습니다.

예를 들어, 제조 공장에서 제품 사진을 약 5000장을 모았고 그 중에서 정상은 3000장, 비정상은 2000장이 있다고 해보겠습니다. 이 데이터를 이용해서 정상은 정상으로 분류하고, 비정상은 비정상으로 분류하도록 잘 학습시킨다면, 새로운 제품 이미지가 입력으로 들어올 때 해당 제품이 불량인지 아닌지를 탐지할 수 있습니다.

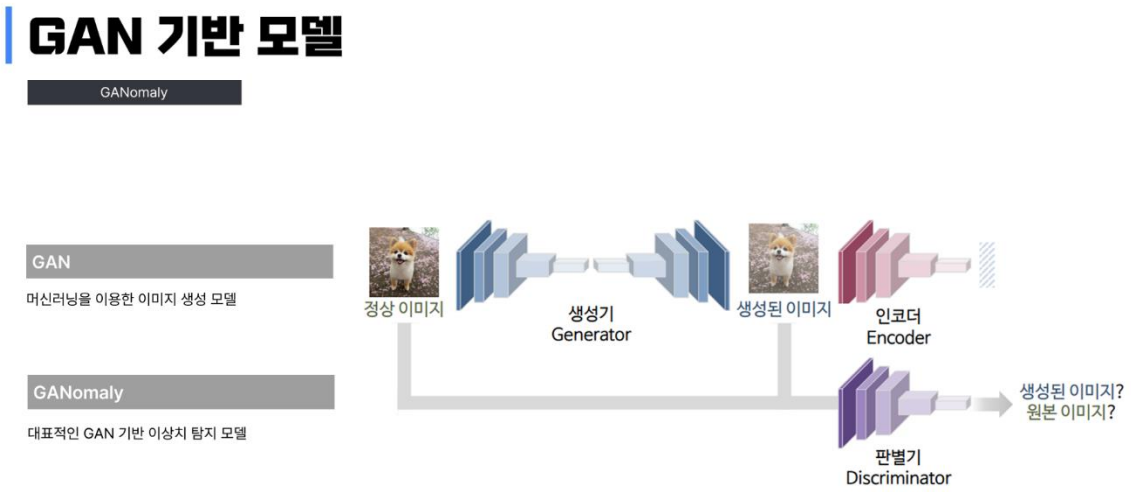
그리고 의료 진단 데이터로 LAG라는 것을 이용하면 녹내장 사진을 비정상으로 취급함으로써 녹내장도 구별할 수 있습니다.

또한, 이상치 탐지는 보안에서 치안을 목적으로도 많이 활용이 됩니다. 대표적으로 보행자 도로를 찍은 CCTV 데이터셋인 UCSD가 있습니다. 보행자 도로이기 때문에 차가 오거나 자전거를 탄 사람이 있으면 해당 프레임을 비정상으로 간주를 할 수가 있는 것입니다. 따라서 걷는 사람들만 있는 프레임을 정상, 나머지를 비정상으로 학습시키면 새로운 프레임이 오더라도 이상 프레임인지를 탐지할 수 있습니다.

이러한 이상치 탐지를 하는 방식은 크게 세 가지로 구성이 됩니다. 먼저 분류 방식은 데이터 각각에 대한 특징 벡터를 파악한 뒤, 정상 특징 벡터들은 원 안쪽에 위치하고, 비정상 특징 벡터들은 원 바깥쪽에 위치하도록 하는 원을 학습시키는 방식입니다. 그런데 특징들은 사실상 고차원상에 위치하기 때문에 원이 아니라 초구를 학습시킨다고 표현합니다.

두 번째는 특징 매칭 방식입니다. 이 방식은 데이터가 정규 분포를 따른다고 가정하고, 실제 특징들의 분포를 추정하여 정규분포의 바깥쪽에 위치한 경우에는 비정상으로 결정하는 이상치 탐지 알고리즘입니다.

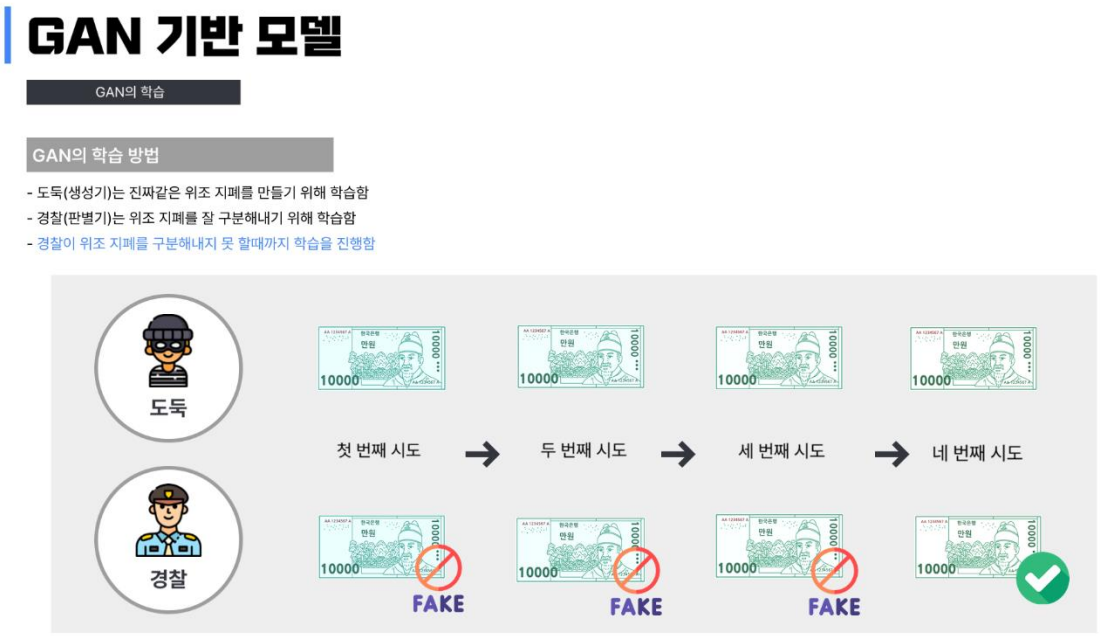
세 번째는 저희가 중심적으로 소개할 재구축 방식입니다. 첫 번째와 두 번째 방식은 정상 입력과 비정상 입력을 모두 이용해서 학습을 시켰다면, 재구축 방식은 정상 데이터만을 이용해서 재구축하는 학습을 진행합니다. 이 방식에서 대표적인 모델인 GANomaly에 대해 말씀드리겠습니다.



GANomaly는 대표적인 GAN 기반 이상치 탐지 모델이며, GAN은 머신러닝을 이용한 이미지 생성 모델입니다. 즉 이미지를 생성해내는 기법을 응용해서 이상치 탐지를 수행하겠다고 생각하시면 됩니다. 이 모델은 크게 생성기, 인코더, 판별기로 이루어져 있습니다. 각 모델을 예시를 통해 알아보겠습니다.

GANomaly에서 사용되는 생성기는 위조 지폐를 생성하는 도둑의 역할을 하게 됩니다.
도둑은 자신에게 들어온 지폐를 분석하고 지폐의 특징을 파악합니다. 그리고 파악한 특징을 이용해서 위조 지폐를 생성하게 됩니다.

GANomaly에서 사용되는 판별기는 지폐가 진짜인지 가짜인지 판별하는 경찰의 역할을 하게 됩니다. 즉, 경찰은 위조지폐와 정상 지폐를 비교 분석하여 위조 지폐를 판별합니다.

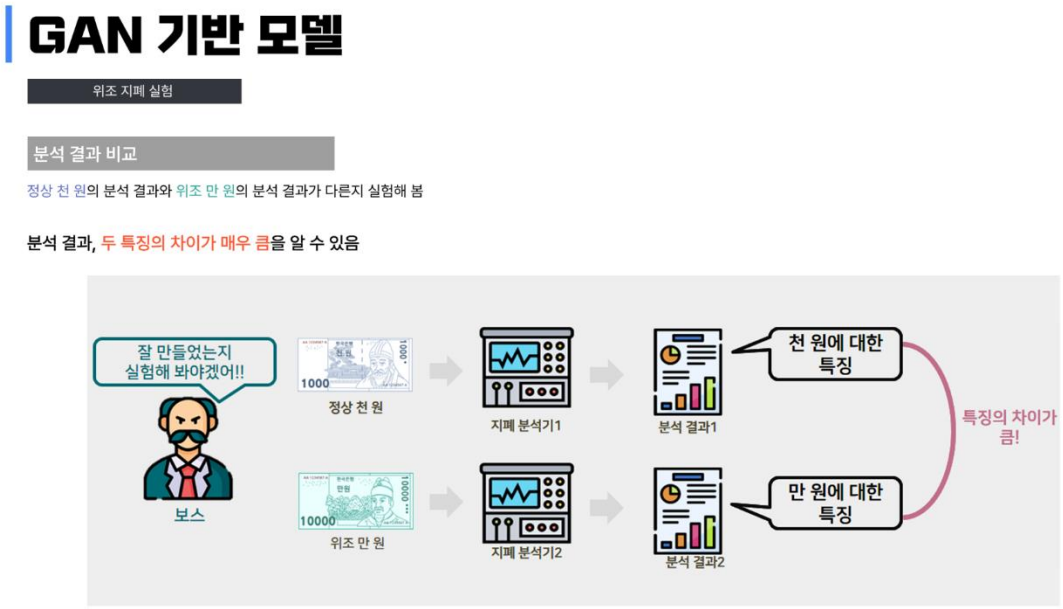


도둑은 진짜 같은 위조 지폐를 만들기 위해 학습하고, 경찰은 위조 지폐를 잘 구분해내기 위해 학습을 진행합니다. 이 둘은 경찰이 위조 지폐를 구분해내지 못할 때까지 학습을 진행하게 됩니다.

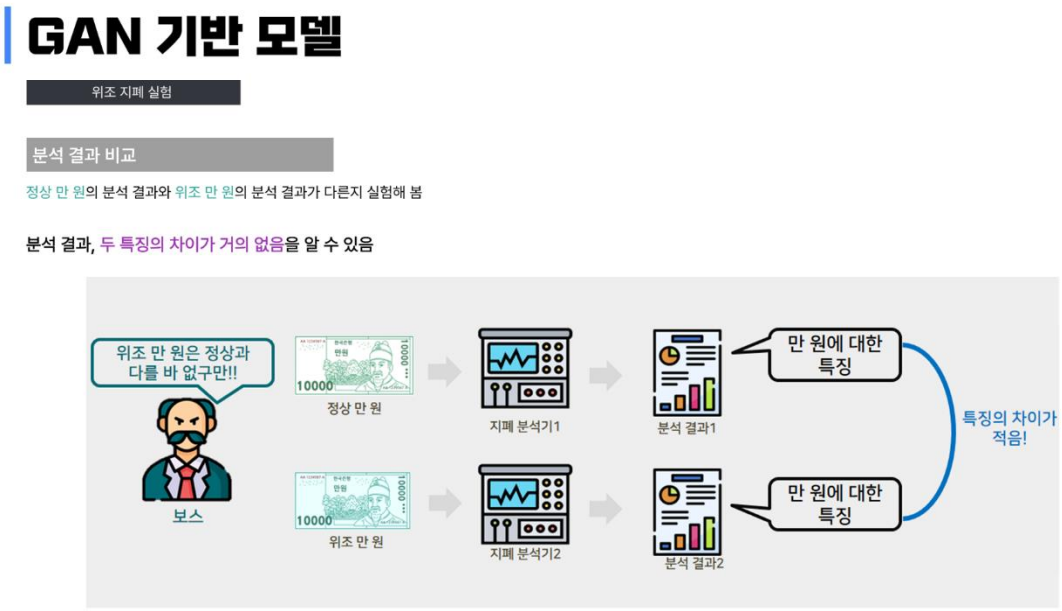
이 때 만 원만 위조할 수 있었던 도둑에게 천 원을 위조하라는 보스의 요구가 있다면 어떻게 될까요?

도둑은 지폐의 특징을 이용해서 위조 만원을 생성하는 일은 잘합니다. 하지만 위조 만 원 지폐만 생성하도록 학습을 했으므로 정상 지폐 천원을 위조 만원으로 만들게 됩니다

그런데 보스는 천 원으로 위조 만 원을 생성해낸 도둑을 당연히 믿지 못합니다.



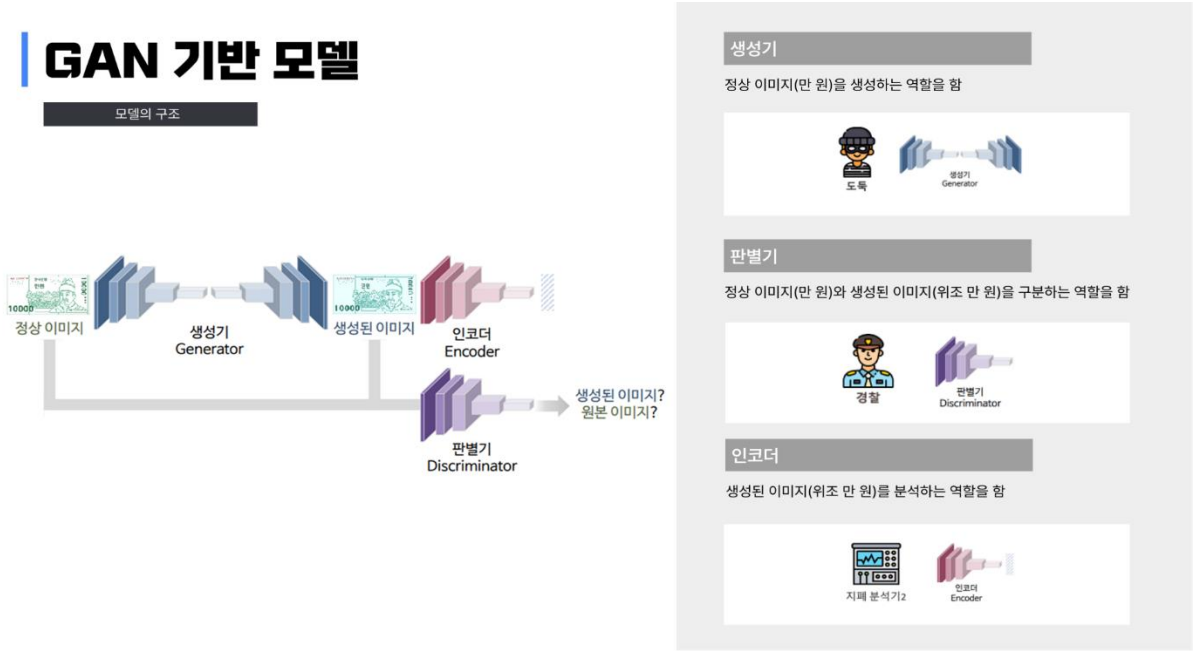
따라서 보스는 도둑이 위조 지폐를 잘 만들었는지 확인하기 위해 정상 천원과 위조된 만원을 지폐 분석기에 넣어 차이점을 알아보는 실험을 해보았습니다. 분석 결과, 두 지폐의 특징 차이가 매우 크게 나왔습니다.



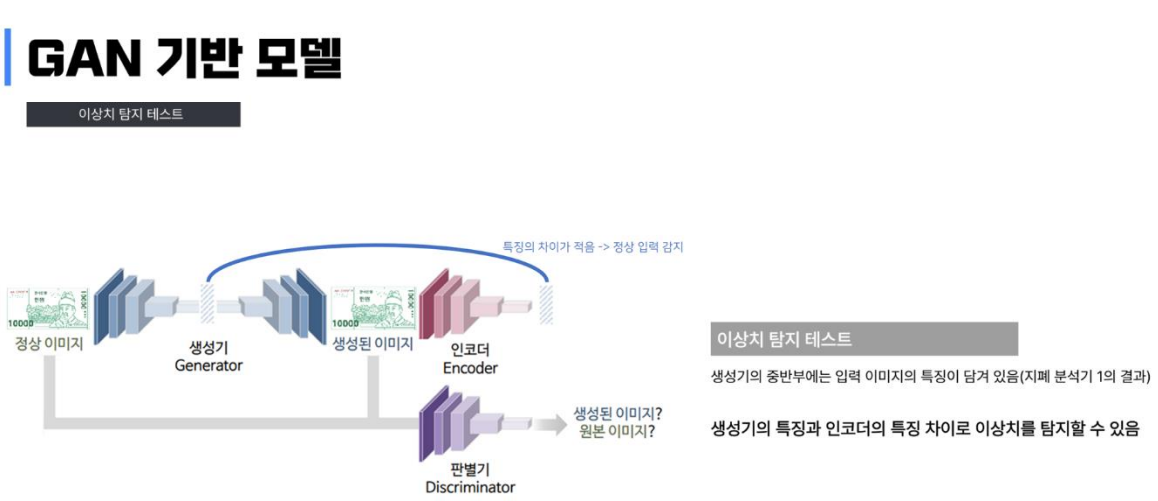
이번에는 정상 만 원과 위조 만 원을 지폐 분석기에 넣어 차이점을 알아보는 실험을 해보았습니다. 분석 결과, 두 지폐의 특징 차이가 거의 없었습니다.

분석 결과1과 분석 결과2의 차이가 이상치 탐지의 핵심 아이디어입니다. 천 원을 비정상, 만 원을 정상이라고 간주하겠습니다. 모델의 입력에 비정상인 천 원이 들어오면 생성자인 도둑에 의해 만 원이 생성되게 됩니다. 생성된 만원을 지폐 분석기2에 넣게 되면 분석 결과1과 분석 결과2의 차이는 크게 발생합니다. 따라서 비정상인 1,000원이 입력으로 들어왔다고 예측할 수 있습니다.

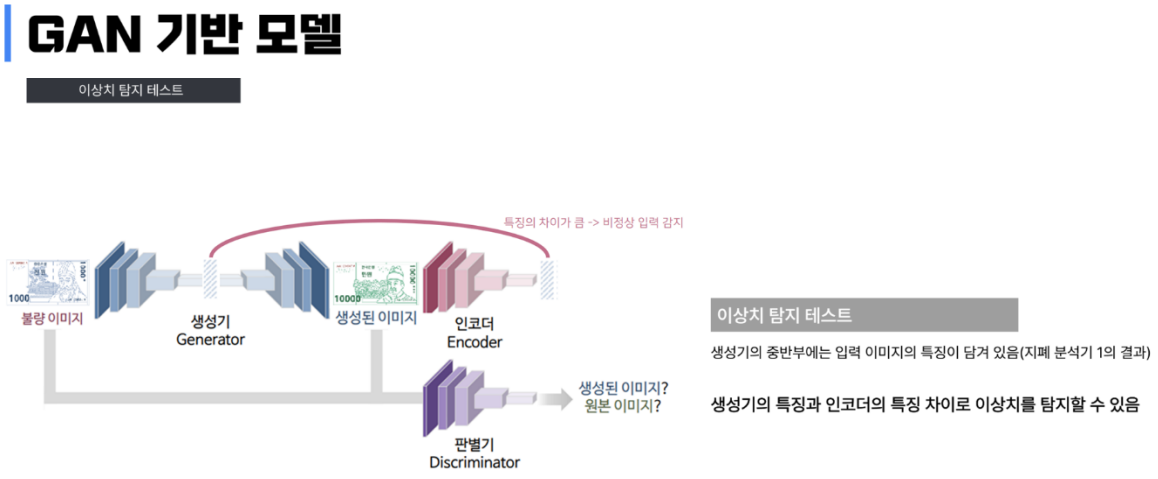
이번에는 모델의 입력으로 정상인 만 원을 넣어보겠습니다. 그럼 생성자인 도둑에 의해 만 원이 생성되게 됩니다. 생성된 만원을 지폐 분석기2에 넣게 되면 분석 결과1과 분석 결과2의 차이는 적게 발생합니다. 따라서 정상인 10,000원이 입력으로 들어왔다고 예측할 수 있습니다.



비유를 통해 살펴봤으니, 이제 원래 모델의 구조로 돌아가겠습니다.
생성기는 정상 이미지를 생성하는 역할이며, 도둑이 만 원을 생성한다고 생각할 수 있습니다.
판별기는 정상 이미지와 생성된 이미지를 구분하는 역할이며, 경찰이 정상 만 원과 위조 만 원을 구분하는 것이라고 생각할 수 있습니다.
인코더는 생성된 이미지를 분석하는 역할이며, 두 번째로 사용된 지폐 분석기라고 생각할 수 있습니다.



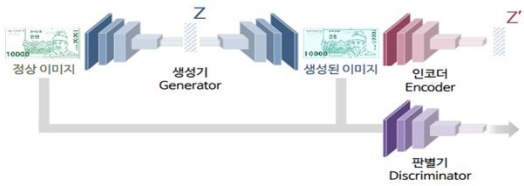
이 때 생성기의 중반부에는 입력 이미지의 특징이 담겨 있습니다. 즉 지폐 분석기1의 결과라고 생각하시면 됩니다. 그리고 인코더의 결과는 생성된 이미지의 특징이 담겨 있습니다. 즉 지폐 분석기2의 결과라고 생각하시면 됩니다. 이 두 결과를 이용해서 이상치 탐지 테스트를 할 때는, 생성기의 특징과 인코더의 특징 차이를 이용하는 것입니다. 즉 특징의 차이가 적으면 정상 입력이라고 감지합니다.



그렇지 않고 특징의 차이가 크면 비정상 입력 혹은 불량 입력이라고 감지합니다.

GAN 기반 모델

이상치 점수 계산



$$ERROR = \frac{1}{n}\sum_{i=0}^{n}(z_i - z'_i)^2$$

$$Anomaly\ Score = \frac{ERROR - ERROR_{min}}{ERROR_{max} - ERROR_{min}}$$

n: $Feature\ Dimension$ (특징의 차원 수)
$ERROR_{max}$: ERROR 값 중 최대값
$ERROR_{min}$: ERROR 값 중 최소값

ERROR

생성기의 특징(z)과 인코더의 특징(z')의 평균 제곱오차
0에 가까울수록 특징이 비슷함

Anomaly Score

ERROR를 정규화해서 0과1사이의 수치로 나타낸 점수
0에 가까울수록 정상

그럼 이 때 수치화한 이상치 점수는 어떻게 계산할까요? 바로 평균제곱오차를 이용합니다. 생성기의 특징을 z라고 하고, 인코더의 특징을 z′라고 할 때, 각각의 특징은 크기가 n인 벡터가 됩니다. 따라서 벡터 요소의 차이 제곱 평균이 작으면 두 벡터가 비슷하다고 표현할 수 있습니다.
이렇게 나온 값을 ERROR라고 하며, 이러한 ERROR는 입력 개수만큼 계산이 됩니다. 따라서 여러 ERROR에 대한 최솟값과 최댓값을 계산할 수 있는데, ERROR에 최솟값을 빼고 (최댓값-최솟값)을 나누는 방식으로 정규화를 함으로써 이상치 점수를 0과 1사이로 나타내는 것입니다.

모델 활용

모델의 활용

GANomaly를 이용한 생체 인증

정상 이미지로 간주한 만 원을 사용자의 손금으로 대체
불량 이미지로 간주한 천 원을 기타 모든 이미지로 대체

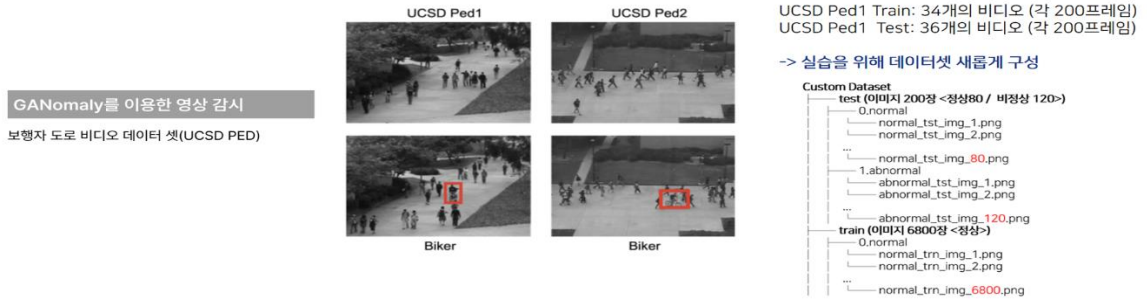
사용자의 손금만으로 학습이 되므로 기타 모든 이미지는 재구축 성능이 낮음 > 비정상

이번에는 모델을 활용하는 방식에 대해 말씀드리겠습니다. 먼저 생체 인증 중 손금 인증을 한다고 하면, 정상 이미지를 사용자의 손금으로 설정하고, 불량 이미지를 다른 사용자의 손금이나 기타 이미지로 설정한 뒤 학습을 한다면, 사용자의 손금으로만 학습이 되므로 나머지 이미지는 재구축 성능이 낮고 비정상임을 확인할 수 있습니다.

모델 활용

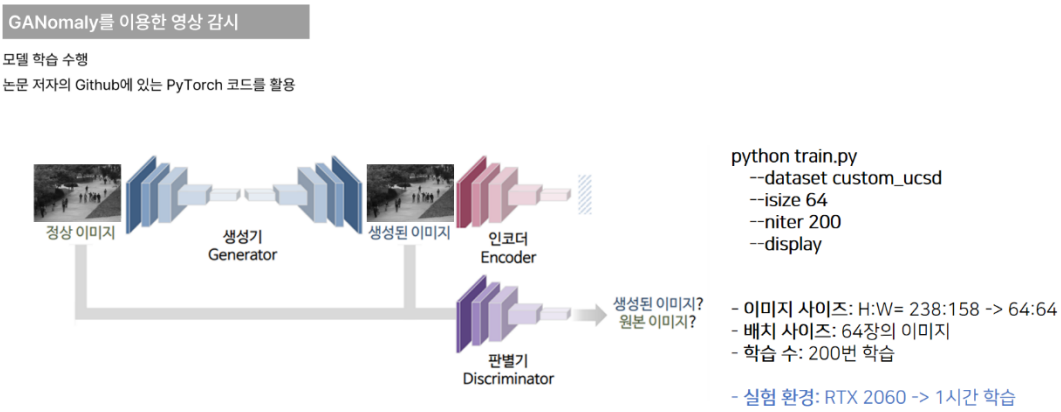
다음은 영상 감시에 대한 예시입니다. 보행자 도로 CCTV의 프레임들이 있을 때, 보행자만 지나다니는 프레임을 정상 프레임으로 설정하고, 자전거를 탄 사람이 있는 프레임이나 기타 프레임을 불량 프레임으로 설정하고 학습을 하면, 정상 프레임만으로 학습이 되므로 나머지 프레임은 재구축 성능이 낮고 비정상임을 확인할 수 있습니다.

실습



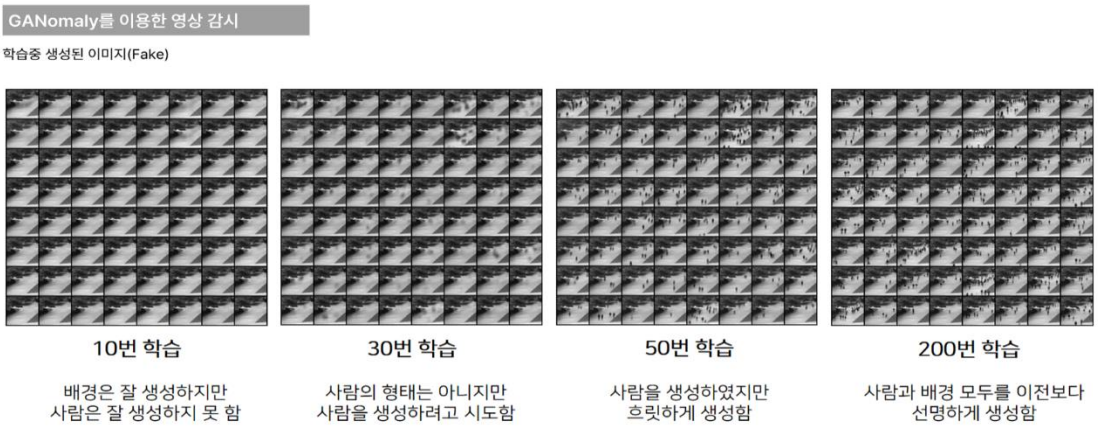
이제 영상 감시에 대한 실습을 해보겠습니다. 데이터셋은 보행자 도로 비디오 데이터셋인 UCSD PED를 사용하였으며, 이 데이터셋은 PED1과 PED2라는 디렉토리로 이루어져 있습니다. 저희는 이 중에서 PED1만을 이용하였습니다. PED1데이터는 다시 훈련 비디오 34개, 테스트 비디오 36개로 이루어져 있으며, 각 비디오는 200프레임으로 저장되어 있습니다. 또한 훈련 비디오에는 보행자만 돌아다니는 정상 프레임만으로 이루어져 있으며, 테스트 비디오에는 정상 프레임과 비정상 프레임을 모두 포함하고 있습니다. 그런데 이 테스트 비디오에서 어떤 프레임이 정상이고, 어떤 프레임이 비정상인지의 라벨링된 정보를 가지고 있지 않기 때문에, 저는 그냥 한 개의 비디오만을 이용해서 정상 프레임과 비정상 프레임을 직접 나눔으로써 테스트 데이터셋을 새롭게 구성하였습니다.

실습



먼저 GANomaly 모델을 학습할 때는 논문 저자의 GitHub에 있는 PyTorch 코드를 활용하였습니다. 이 때 모델의 입력으로 사용되는 이미지 사이즈는 관습에 따라 정사각형 형태로 설정하였고, 한 번에 한 개의 이미지만 입력으로 주는 것이 아니라 64장의 이미지를 입력으로 주었습니다. 또한 학습 데이터 6800장을 생성하는 것을 한 번 학습한다고 표현하는데, 이러한 과정을 200번 수행하였습니다. 실험은 저희 집 컴퓨터의 GPU를 이용하였으며, 약 1시간이 소요되었습니다.

실습

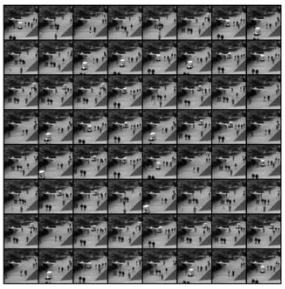


다음은 학습중 생성된 이미지들을 시각화한 자료입니다. 10번 학습을 할 때는 배경은 잘 생성하지만 사람을 잘 생성하지 못하였고, 30번 학습을 할 때는 사람의 형태는 아니지만 사람을 생성하려고 시도하였습니다. 50번 학습을 할 때는 사람을 생성하였지만 흐릿하게 생성하였고, 200번 학습을 할 때는 사람과 배경 모두를 이전보다 선명하게 생성하였습니다.

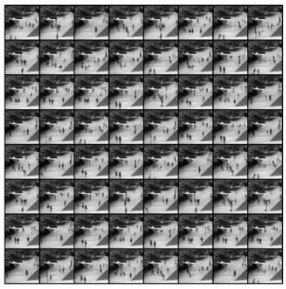
실습

GANomaly를 이용한 영상 감시

테스트



입력 영상들
(Real)



생성 영상들
(Fake)

정상 영상은 잘 생성하지만,

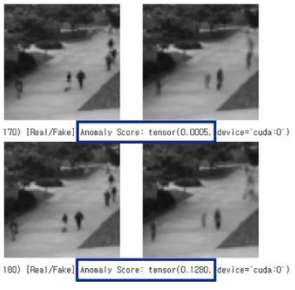
비정상 영상(차를 포함하는 영상)은 생성하지 못함

다음은 테스트입니다. 테스트에는 말씀드렸다싶이 정상 프레임과 비정상 프레임이 모두 포함돼있기 때문에, 자세히 보시면 차가 돌아다니는 이미지와 보행자만 돌아다니는 이미지가 입력 영상에 섞여있는 것을 확인할 수 있습니다. 그러나 정상 영상으로만 학습을 했기 때문에 비정상 영상, 즉 차를 포함하는 영상은 생성하지 못하는 것을 확인할 수 있습니다.

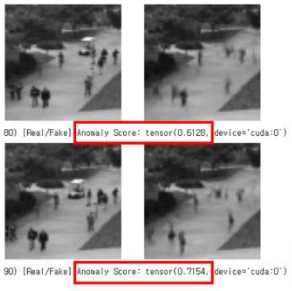
실습

GANomaly를 이용한 영상 감시

이상치 점수 확인



정상 영상
(Normal)



비정상 영상
(Anomaly)

정상 영상은 Anomaly Score가 낮지만(0.0005),

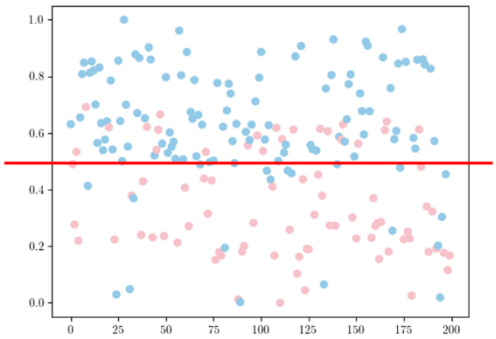
비정상 영상은 Anomaly Score가 높음(0.7154)

그리고 정상 영상과 비정상 영상의 이상치 점수도 확인해보았습니다. 정상 영상인 경우 0.0005로 매우 낮은 수치가 나왔고, 비정상 영상인 경우 0.7154로 높은 수치가 나왔습니다.

실습

GANomaly를 이용한 영상 감시

이상치 점수의 산점도



하늘색 점은 비정상 입력에 대한 이상치 점수(Anomaly Score)

분홍색 점은 정상 입력에 대한 이상치 점수(Anomaly Score)

Threshold(임계점)가 0.5일 때 정상과 비정상을 대략적으로 구분할 수 있음

$$ERROR = \frac{1}{n}\sum_{i=0}^{n}(z_i - z'_i)^2$$

$$Anomaly\ Score = \frac{ERROR - ERROR_{min}}{ERROR_{max} - ERROR_{min}}$$

이러한 이상치 점수들을 산점도로 나타내보았습니다. 하늘색 점은 비정상 입력에 대한 이상치 점수이고, 분홍색 점은 정상 입력에 대한 이상치 점수입니다. 산점도를 보았을 때 Threshold가 약 0.5일 때 정상과 비정상을 대략적으로 구분할 수 있는 것을 확인할 수 있습니다.

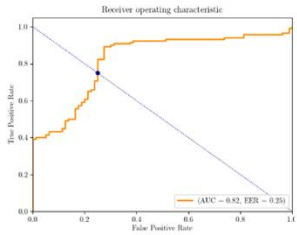
실습

GANomaly를 이용한 영상 감시

시뮬레이션(Threshold = 0.5)

UCSD PED1의 24번째 영상으로 시뮬레이션

차가 지나다니는 비정상 프레임이 왔을 때, 이상치 탐지를 잘 수행함



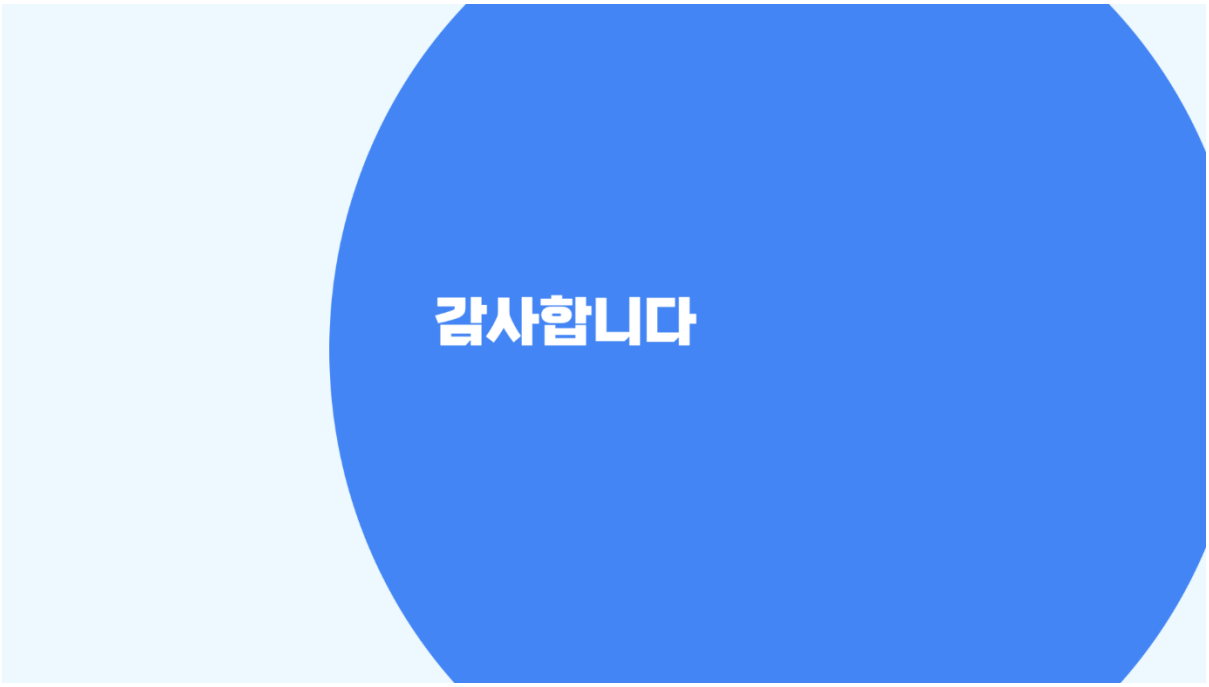
AUC: 딥러닝 분류 모델의 평가 척도, 1에 가까울수록 분류를 잘 수행함

마지막으로 시뮬레이션을 해보겠습니다. 테스트 데이터로 사용된 UCSD의 24번째 영상을 가져왔습니다. (비디오 영상 클릭)

100% 정확하지는 않지만 차가 지나다니면 Anomaly가 뜨고, 차가 사라지면 Anomaly가 사라지는 것을 확인할 수 있습니다.

그리고 AUC라는 딥러닝 모델 평가 척도를 이용해서 점수를 확인해본 결과 82점이 나왔습니다.

실습 코드는 제가 깃허브에 올렸으니 링크나 QR코드를 이용해주시면 될 것 같습니다.



이상으로 발표를 마치도록 하겠습니다. 감사합니다.