



2018년 8월 석사학위 논문

Conditional Generative Adversarial Networks를 이용한 물방울 왜곡 복구

조선대학교 산업기술융합대학원

소프트웨어융합공학과

구 수 한









Conditional Generative Adversarial Networks를 이용한 물방울 왜곡 복구

Restoring water drop on window using Conditional Generative Adversarial Networks

2018년 8월 24일

조선대학교 산업기술융합대학원

소프트웨어융합공학과

구 수 한





Conditional Generative Adversarial Networks를 이용한 물방울 왜곡 복구

지도교수 양 희 덕

이 논문을 공학석사학위신청 논문으로 제출함.

2018년 4월

조선대학교 산업기술융합대학원

소프트웨어융합공학과

구 수 한





구수한의 석사학위논문을 인준함



2018년 5월

조선대학교 산업기술융합대학원



, i *



목 차

ABSTRACT

제1장. 서론
제1절. 연구 배경 및 목적
제2절. 연구 내용 및 구성
제2장. 관련 연구
제1절. 물방울 및 먼지왜곡 감지와 제거방법4
제2절. GAN7
제3장. CGAN을 이용한 영상 추론
제1절. 건너뛰기 연결을 포함한 생성기
제2절. 패치로 설계된 판별기
제4장. 실험 결과 및 분석
제1절. 실험 환경
제2절. 결과 및 분석
제5장. 결론
참고문헌





표 목 차

[표 4-1] 학습 및 시험을	위한 환경	17
[표4-2] 복원된 영상들의	PSNR ·····	26





그림목차

[그림 1-1] 물방울 왜곡의 종류에 따른 원영상과 왜곡 영상
[그림 2-1] GAN의 기본 원리
[그림 3-1] 인코더-디코더 구조
[그림 3-2] 건너뛰기 연결이 포함된 생성기의 구조
[그림 3-3] 생성기의 학습구조
[그림 3-4] 판별기의 학습구조
[그림 4-1] 학습 및 시험을 위해 혼합에 사용된 영상 및 잔상이 남은 영상18
[그림 4-2] 물방울 왜곡 영상과 필터로 복원한 영상의 잘못된 예
[그림 4-3] 일반적인 물방울 왜곡에 대한 복원 영상
[그림 4-4] 물줄기 및 물안개 왜곡에 대한 복원 영상
[그림 4-5] 물방울 왜곡이 심한 왜곡 영상에 대한 복원 영상
[그림 4-6] 물줄기가 포함된 왜곡 영상에 대한 복원 영상
[그림 4-7] 짙은 물안개가 포함된 왜곡 영상에 대한 복원 영상
[그림 4-8] 분류에 따른 왜곡별 복원 영상의 PSNR
[그림 4-9] 학습 횟수에 따른 CGAN의 손실률





ABSTRACT

Restoring water drop on window using Conditional Generative Adversarial Networks

Suhan Koo Advisor : Prof. Heedeok Yang Ph.D Department of Computer Engineering Graduate School of Chosun University

Due to the recent development of artificial intelligence technology, self-driving cars and smart cars are drawing attention. These technologies get many problems from development and application, and the biggest one is safety. In case of self-driving cars or smart cars, camera lenses are used to recognize obstacles and objects around them. Current technologies are used for driving safety purposes, other uses of camera lenses are preventing crime, but if they start with CCTV (Closed-circuit televisional), they can also cause a variety of signals and speed controls, as well as station-stop violations.

The restoration of a polka-down distorted image is divided into the detection of water droplets and the image restoration. Using the properties of water droplets to detect them and then replace the detected area or restore them using conventional approach. However, it is difficult to completely restore the image, without the demage of resolution or quality. Removing water droplets from the video is a difficult but very important task at the same time.





For the purpose of solving this problem, this study applies a new methodology to the Generative Adversarial Networks (GAN), to eliminate water droplets distortion it would be better to generate new images which are similar original images, rather than restoring the original images, So that the quality of the marine images restored by GAN will be excellent. However, since the quality of images generated by the existing GAN is not good and can not be created separately from the desired parts, CGAN is applied for the proposed method.

The evaluation is performed by comparing the PSNR. Values of the original images with the restored images using each filter and those generated using GAN.





제1장 서론

제1절 연구 배경 및 목적

최근 인공지능 기술의 발전으로 자율주행 자동차가 주목 받고 있다. 이와 같은 기술 들은 개발되어 적용하기까지 많은 문제를 겪는데, 가장 중요하게 여기는 건 안전사고 예방이다. 자율주행 자동차의 경우 주변의 장애물이나 객체를 인식할 때 카메라 렌즈 를 이용하는데, 비오는 날이나 렌즈에 이물질이 묻게 되면 인식률이 크게 떨어지기 때 문에 안전사고 문제로 이어져 렌즈를 통한 객체 인식기술의 향상이 중요하다. 현재 차 량에 적용된 기술로는 주행안전의 용도로 사용되는 전·후방 충돌 경고, 신호등 또는 주변 환경인식을 위한 전용 카메라의 경우 창문이나 카메라 렌즈에 물방울이 떨어지거 나 비오는 날씨가 되면 그림 1-1 (b)와 같이 카메라 이미지에 왜곡이 발생하게 된다. 범죄예방을 위해 신호 및 과속단속, 주·정차위반 단속 등의 목적으로 사용되는 CCTV(Closed-circuit television) 또한 왜곡이 일어나면 객체의 판단이 어렵다.

물방울 왜곡 영상에 대한 복원은 물방울 검출과 영상 복원으로 구분된다. 먼저, 물 방울의 여러 특성을 이용하여 물방울들을 검출한 후 검출된 영역을 교체하거나 다른 방법을 통해 복원하는 것이 기존 연구의 방식이다. 이러한 방법들의 경우 영상에 대한 완벽한 복원이 힘들고, 복원 된 영상은 화질이 좋지 않다는 단점을 가지고 있다. 이처 럼 영상에서 물방울을 제거하는 것은 까다롭지만 중요한 작업이다.

이러한 문제를 개선하기 위해 본 연구에서는 물방울 왜곡을 제거하고자 기존의 연구 들과는 다르게 새로운 방법인 GAN(Generative Adversarial Networks)[1]를 접목한다. 원래의 영상을 복원하기 보다는 원래의 영상과 같은 새로운 영상의 생성이 가능하다면 본 연구의 방법을 통해 복원된 영상의 화질은 우수 할 것이다. 그러나 기존의 GAN을 통해 생성되어지는 영상의 화질은 양호한 편이 아니고 원하는 부분만 따로 생성할 수 가 없기 때문에 단점을 보완하고자 CGAN(Conditional Generative Adversarial





Networks)[2]를 이용해 영상으로부터 물방울 왜곡을 제거한다.

본 연구에서는 다양한 왜곡에서의 복원을 위해 그림 1-1과 같이 왜곡 영상을 학습시 켜 영상을 크게 5가지 종류로 분류하였고, 차례대로 일반적인 물방울 왜곡, 물줄기 및 물안개를 포함한 왜곡, 다량의 물이 렌즈에 묻어 일어난 왜곡, 물줄기가 포함된 왜곡, 짙은 물안개가 포함된 왜곡으로 분류하여 영상을 생성하였다. 생성된 영상의 비교 평 가를 위해 왜곡이 일어나기 전의 영상과 필터를 이용해 복원한 영상들 그리고 cycle GAN을 이용해 복원된 영상의 손실정보를 평가할 때 사용되는 최대 신호 대 잡음비인 PSNR(Peak Signal-to-noise ratio)을 이용한다. 따라서 본 연구에서도 원영상과 각 필 터를 이용해 복원된 영상과 및 GAN을 이용해 생성된 영상의 PSNR 값으로 평가를 수행 한다.



(a) 원영상



(b)물방울왜곡이일어난영상 그림 1 -1 원영상과 물방울 왜곡이 일어난 영상의 예





제2절 연구 내용 및 구성

저해상도의 영상을 고해상도로 변환하거나 현실감 있는 새로운 영상을 생성, 탐지 분류기에 접합되어 성능이 좋은 분류기를 만드는 등의 목적으로 GAN은 다양하게 사용 된다. 본 연구에서는 기존 GAN의 판별기와 생성기에게 어떤 추가정보를 조건부로 제공 하여 원하는 부분만 생성 가능한 CGAN을 이용한다.

CGAN의 장점으로 설명하자면 생성되는 데이터를 마음대로 조절 할 수 있다는 점이 다. 이러한 CGAN의 장점을 살려 본 연구에서는 물방울 로 인한 왜곡이 일어난 사진을 원래대로 복원하는 것을 제안한다.

본장인 서론에 이어 2장 관련 연구에서는 본 연구의 이론적인 배경과 관련하여 CGAN 의 설명에 앞서 이동 중인 차량에 설치된 카메라에서 물방울 왜곡이나 먼지가 묻은 창 문 사진 복원 등을 살펴보고 GAN에 대해 설명을 하여 본 연구 내용의 이해를 돕는다.

3장에서는 CGAN을 이용한 영상추론방법에 대하여 기술한다. 본 논문에서 영상을 생성하기 위해 사용된 건너뛰기 연결이 포함된 생성기와 판별기에 대해 소개한다.

4장에서는 실험환경과 CGAN의 학습을 위한 영상 생성 과정에 대하여 언급하고, 각 필터로 복원된 영상들과 cycle GAN과 CGAN으로 생성된 영상에 대한 PSNR을 구해 비교 평가를 실시하였다.

마지막으로 5장에서는 본 연구에 대한 전체적인 결론으로 마무리한다.





제2장 관련 연구

제1절 물방울 및 먼지 왜곡 감지 및 제거방법

1.물방울 제거

Hsien-Chou Liao 등이 제안한 방법은 물방울 검출 및 물방울 제거로 구성되는데 각 단계는 다음과 같은 특징으로 구성된다[3]. 첫 단계는 물방울 검출 단계로 물방울 특 성에 따라 설계되었고, 두 번째 물방울 제거 단계에서는 제거 작업은 크게 두 가지로 이루어진다. 하나는 이전 프레임의 모든 물방울에 대한 패치 위치이고, 다른 하나는 모든 물방울 영역의 교체이다. 템플리트 매칭을 사용하여 패치를 찾고 두 이미지의 스 티칭과 유사한 방법을 사용하여 도로 지역의 물방울을 제거하였다.

David Eigen 등의 물방울 왜곡 제거의 경우 물방울 왜곡의 데이터를 가지고 있지 않 기 때문에, 여러 장면의 영상을 촬영하여 학습 세트를 만들었다[4]. 물방울 왜곡 영상 의 경우 M_gF_2 코팅이 된 유리에 물을 분사하여 하여 실제 비와 매우 유사한 방울을 만들도록 주의를 기울였다. M_gF_2 코팅은 무반사 코팅의 한 종류로써 보통의 유리들 보다 투과율 증대 및 피사체가 역으로 반사되는 효과를 최대한 줄임으로 효율을 크게 높여주는 코팅 선택사항이다. 깨끗한 영상과 물방울 왜곡 영상의 차이를 제한하기 위 해 모든 영상에 정적인 개체만 포함시켰다. 물방울 제거를 위하여 먼지 제거와 마찬가 지로 선형 최소 제곱 시스템을 풀어 미세한 디테일과 모서리 부분을 적절히 보존하면 서 대부분의 물방울을 제거할 수 있었다.

Martin Roser 등은 빗방울 검출을 위해 영상에서 관심지역을 추출하고, 각 영역에 인공 빗방울 패턴을 렌더링 하는 측광 빗방울 모델을 사용하여 검출하였다[5]. 검출된 빗방울을 검증하기 위해 상관계수의 강도와 그라디언트를 1차 미분을 통해 최대화 한



다. 검출된 빗방울에 덮인 이미지 영역을 재구성하는 것은 근처 프레임의 강도 정보를 이용하여 가능하다. 단순한 영상 병합은 시각화, 객체 그림자 등의 오류가 발생하기 때문에 멀티 레벨 스플 라인[6]을 사용하여 효과적으로 제거하였다.

Kshitiz Garg 등은 비의 역학적 특성과 광도 측정 모델을 바탕으로 비디오에서 비가 내리는 지역을 탐지하기 위한 알고리즘을 설계하였다[7]. 비의 역학적인 특성으로 빗 방울을 검출한 뒤 광도 측정 모델을 이용하여 제거를 하였으나 빗방울 감소가 현저하 게 이루어져 역학 모델을 이용해 빗방울 제거를 한 번 더 실행하여 비디오 프레임에 적용하여 빗방울을 제거한다.

Peter Barnum 등은 비나 눈을 검출하기 위해 물리적 통계적 특성을 바탕으로 주파수 공간에 대한 모델[8]을 개발하여 비나 눈을 검출하고 제거를 위해 노치 필터링[9]과 유사한 방법을 사용하여 제거하였으나 원하지 않는 부분까지 제거가 되어 주파수의 예 상 값과 실제 값의 비율로 추정하여 제거를 수행하였다. 이는 개별 프레임에 대해 수 행하는 것이지만 비가 내리는 것은 동적이기 때문에 프레임을 정렬하여 이미지에서 강 하게 보이는 비와 눈을 찾아 제거하여 다른 프레임과 비교 후 계산을 하면서 제거하였 다.

Xiaopeng Zhang 등은 영상에서 나타나는 물방울의 형태학적인 특성과 광도를 이용하 여 빗방울을 검출하였다[10]. 영상의 각 픽셀에 대해 전체 비디오에 대한 조도가 수집 되어 히스토그램을 계산하여 해당 연구에서 소개한 클러스터링을 적용한 후 픽셀별 조 도의 크기가 큰 경우 다른 픽셀의 가장 작은 값과 평균값을 적용해 가며 이미지 복원 을 하였고 흐려진 부분은 알파혼합을 통해 제거하였다.

Collection @ chosun



2.먼지 제거

David Eigen 등은[4] 먼지를 제거하는 방법을 학습하기 위해, 이미지에 먼지를 합성 하여 투명하고 투명한 이미지 쌍을 생성했다. 합성에 사용된 방법은 알파 혼합이며 식 은 다음과 같다.

$$I' = p\alpha D + (1 - \alpha)I \tag{1}$$

여기서, *I*와 *I*'은 각각 잡음이 없는 영상과 잡음이 추가된 영상이다. *p*는 RGB 각각에 대한 균일 분포이고, 0과 1사이의 다른 난수가 곱해져 밝기가 변한다. α는 이미지와 같은 크기의 투명 마스크이며, *D*는 이미지와 동일한 크기의 먼지 성분이다. 각 픽셀 에서 α와 α*D*에 대한 선형 최소 제곱 시스템을 풀어 먼지 제거를 수행하였다.

또한 이 연구에서는 제안되어진 네트워크가 정확한 패턴을 암기하고 일치시키지 못 하게 하고자 실제 먼지가 묻은 영상도 준비하였고 실제 먼지가 묻은 영상과 인위적으 로 합성한 영상의 복원을 실험하였다. 비교평가로는 원래의 목표 영상과 만들어진 영 상의 PSNR(Peak Signal-to-noise ratio)을 계산하여 수치 값을 구하여 비교 하였다. 비록 학습 데이터 세트는 전부 합성된 먼지로 구성 되었지만 네트워크가 두 경우 모두 잘 수행하였다.

먼지 제거 연구 및 물방울 왜곡 제거를 포함 Hsien-Chou Liao등의 관련연구들을 살 펴본 결과 현재 물방울 왜곡 제거에 CGAN을 접목한 연구가 없어 본 연구에서는 CGAN을 접목하여 물방울 왜곡제거를 연구하고자 한다.





제2절 GAN

GAN은 크게 두 부분으로 나뉘는데 생성기인 G(generator)와 판별기인 D(discriminator)가 그림 2-1은 생성기가 영상 생성을 하여 판별기의 판별 여부에 따라 학습되는 과정으로 판별기와 생성기가 서로의 성능을 개선해 나아간다.



그림 2-1 GAN의 기본 원리 생성기의 생성영상과 목표영상을 입력받아 판별기의 판별여부에 따라 생성기의 학습이 진행됨

GAN에서 G가 거짓 데이터를 생성하게 되면 D는 실제 데이터와 G가 생성한 거짓 데이 터로부터 실제와 거짓을 구분하게 된다. 여기서 G는 D를 속이지 못한 정보를 D는 G로 인해 속은 정보를 입력 받아 학습하고 이 과정을 반복한다. 학습이 반복되면 G는 실제 와 가까운 데이터를 생성하게 되고, D는 점점 데이터의 실제와 거짓을 잘 구분하게 되 는데 충분히 학습된 G를 이미지 생성에 이용한다.





GAN의 이론은 판별기가 실제와 가짜를 구별 할 수 없을 정도 $p_d = 0.5$ 에 이르게 된다 는 것이다. 여기서 p_d 는 D가 데이터를 입력 받았을 때 실제와 거짓을 구분할 확률이 다.

GAN의 생성기 G와 판별기 D에 대한 V(G,D)에 대한 함수는 다음과 같다.

$$\min\max_{Q} V(D,G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim pz(z)} [\log (1 - D(G(z)))]$$

$$G \quad D$$
(2)

여기서, $E_{x \sim p_{data}(x)}$ 의 *E*는 기댓값, *x*는 임의의 데이터, *x*가 따르는 분포는 $x \sim p_{data}(x)$ 로 나타낸다. $E_{x \sim p_{data}(x)}[\log D(x)]$ 의 D(x)는 데이터의 분포로부터 왔을 확률, *z*는 임의 의 노이즈이다. 이 항은 실제 데이터와 G(z)가 만들어낸 데이터에 정확한 라벨을 붙 이기 위해 학습이 되고, $E_{z \sim p_x(z)}[\log(1 - D(G(z)))]$ 의 G(z)는 $\log(1 - G(z))$ 을 최소화하 려는 방향으로 학습되기 때문에 GAN의 목적함수는 min*imax*라는 문제가 된다.

GAN의 경우 여러 문제가 있는데 다음과 같다. 첫째로 생성된 결과가 불안정하였고, 두 번째로는 기존의 다른 기계학습 기법들과 달리 처음부터 끝까지 어떤 형태로 결과 가 나오게 되었는지 그 과정을 알 수 없었다. 마지막으로 GAN의 결과물 자체가 새롭게 만들어진 샘플이기 때문에 기존 샘플과 비교하여 얼마나 비슷한 지 확인할 수 있는 정 량적 척도가 없고, 사람이 판단하더라도 이는 주관적이기 때문에 판단의 근거가 부족 하다[35].

GAN의 단점을 보완하고 이를 이용한 다양한 연구들이 등장하는데 이중 하나인 EnhanceNet[11]은 GAN의 손실함수를 적용해 이미지 해상도를 높이는 기능의 성능을 높 였다. EnhanceNet이 이미지 해상도를 높이는 문제에 GAN구조를 적용한 것은 생성기의 입력으로 임의의 노이즈가 아닌 저해상도의 이미지를 입력하여 판별기가 만들어진 가 짜 고해상도 이미지와 실제 고해상도 이미지를 구분하며 서로 대립해 가며 성능을 높 인다.





현실감 있는 인공데이터를 만들어내는데 사용되어지는 SimGAN[12]은 기존 GAN의 생 성기가 인공데이터를 받아 실제 데이터로 변환하는 역할을 판별기는 생성기가 생성한 가짜와 실제 데이터를 구분하는 과정을 통해 생성기가 학습을 해가며 성능을 높이게 된다.

DCGAN[13]은 GAN의 가장 큰 문제점인 불안정한 구조 때문에 등장하게 되었는데, 단 어 그대로 CNN과 결합하여 구조의 안정성을 추구하고 대부분의 상황에서 학습을 안정 적으로 시킨 것에 큰 성과를 거두었다.

CGAN과 GAN의 구조적인 차이점은 생성기와 판별기 모두에게 약간의 추가정보인 *y*를 조건으로 제공해 조건부 모델로 확장 할 수 있다는 점이다. 여기서 *y*는 클래스 레이 블이나 다른 형태의 데이터와 같은 보조데이터이다. 이는 구조와 학습과정 자체가 간 단하며 장점으로는 사용자가 원하는 부분만 생성이 가능하다는 점이 가장 큰 장점으로 꼽힌다. 이러한 장점으로 인해 영상의 낮을 밤으로 바꾼다던지, 흑백 영상에서 컬러 영상으로 변환 레이블로 구성된 맵을 사진으로 합성한다거나, 스케치로만 이루어진 영 상을 실제 영상으로 만드는 등의 다양한 분야에 우수한 결과를 얻었다.





제3장 CGAN을 이용한 영상 추론 방법

본 연구에서 사용 된 CGAN과 GAN의 차이는 노이즈 벡터 *z*와 임의의 *y*를 결합시켜 가짜 데이터를 생성시키고 원래의 데이터 *x*와 *y*를 결합시켜 판별기의 입력으로 넣는 다.

CGAN의 손실률에 대한 기본 목적 함수는 다음과 같다.

$$L_{cGAN}(G,D) = E_{x,y}[\log D(x,y)] + E_{x,z}[\log (1 - D(x, G(x,z)))]$$
(3)

여기서, $E_{x,y}[\log D(x,y)]$ 는 판별기의 기댓값으로 $\log D(x,y)$ 는 임의의 데이터 x와 y가 결합된 상태에서의 확률 분포이다.수식 (2)와 마찬가지로 D(x,G(x,z))는 $\log D(x,y)$ 에 대하여 최소화 하려는 방향으로 학습한다.

CGAN의 판별기의 조건화 중요성 테스트를 위해 조건화가 되지 않은 판별기와 비교한 다. 판별기는 *x*를 관찰하지 않는다는 조건을 포함해 수식으로 표현하면

$$L_{GAN}(G,D) = E_y[\log D(y)] + E_{x,z}[\log (1 - D(G(x,z)))]$$
(4)

기존의 CGAN에서 생성기의 입력에 *z*로 가우시안 노이즈를 추가하여 사용하여 봤으 나 효과적이지 못하였다. 그러나 다른 연구[14]에서 전통적인 손실함수인 L1, L2를 혼 합하는 것이 더 유익하다는 것을 알았다.

혼합을 하게 되면 판별기의 기능은 변하지 않지만 생성기는 판별기를 속일뿐 아니라 좀 더 실제의 영상에 가까워지는 출력영상을 만들어내는데 L2보다는 L1의 생성영상이 결과가 더 우수하기 때문에 L1 함수를 사용하고 수식으로 표현하면 수식(5)와 같다.

$$L_{L1}(G) = E_{x,y,z}[\| y - G(x,z) \|_{1}]$$
(5)

- 10 -





본 연구의 최종 목표는 다음과 같다[15].

$$G^* = \arg\min\max_{cGAN}(G, D) + \lambda L_{L1}(G)$$

$$G D$$
(6)

본 연구의 생성기와 판별기의 구조는 DCGAN에서 적용 되었고, 생성기와 판별기는 convolution-BatchNorm-ReLu [16]형식을 사용하였다.





제1절 건너뛰기 연결을 포함한 생성기

영상 간 변환 문제를 정의하는 기능은 고해상도의 입력 그리드를 고해상도의 출력 그리드에 매핑 하는 것이라고 할 수 있다. 또한 고려해야 할 사항으로는 입력과 출력 의 외관은 다르지만 동일한 기본 구조의 표현이라는 점이다. 이전의 많은 연구들 ([17],[18],[19],[20],[21])은 인코더-디코더 연결[22]을 이용하였다.

그림 3-1은 인코더-디코더를 이용한 영상전송 과정을 표현하는데 영상이 계층별로 전송되면서 영상의 너비, 높이, 색상채널의 크기가 변함을 확인한다. 인코더-디코더를 이용할 때 생성기는 입력 영상을 받아 인코더를 이용하여 훨씬 작게 축소하려고 시도 한다.



그림 3-1 인코더 디코더의 구조 (계층별로 이동시 영상의 너비, 높이와 색상채널 의 크기가 변함)

인코더 사용 시 각층을 통과 할 때 축소되면서 조절되다가 병목현상이 발생하여 이 과정이 역전되며 병목현상을 포함하며 모든 계층을 통과해 정보를 손실해버리는 문제 가 발생하게 된다. 많은 이미지 변환 문제에 있어, 입·출력 간 많은 양의 정보가 공유 되기 때문에 이와 같은 문제가 발생하면 본 연구에서는 영상 생성이 곤란하다 그래서





정보를 직접적인 연결을 통해 직접 전송하는 것이 바람직하다.

인코더-디코더 연결과 같이 정보 손실을 피하기 위해 그림 3-2와 같이 생성기에 건 너뛰기 연결을 추가하였다. 인코더-디코더와 같이 영상의 크기와 색상채널을 전달하며 계층별로 이동하면서 크기가 변함을 확인할 수 있다. 여기서 건너뛰기 연결은 인코딩-디코딩 부분을 사용하지 않는 경우 이를 우회하는 옵션을 제공하며 정보를 직접적으로 전달하게 된다. 건너뛰기 연결은 모든 채널에서 벡터를 단순히 사슬처럼 연결하는 방 식이다.



그림 3-2 건너뛰기 연결이 포함된 생성기의 구조 (인코더-디코더를 사용하지 않을 경우 영상에 대한 정보를 건너뛰기 연결을 이용해 바로 전달함)

그림 3-3는 생성기의 학습과정이다. 입력영상이 들어오게 되면 생성기를 통해 출력 을 하여 입력-출력영상과 비교를 하고 출력-목표 영상과 비교를 통해 생성기의 가중치 가 조정되며 학습을 계속 반복한다. 생성기의 가중치는 판별기의 출력-목표 영상의 차 이에 따라 조정된다.

Collection @ chosun





그림 3-3 생성기의 학습구조 (판별기를 통해 영상을 비교하여 생성기의 가중치가 조절됨)

판별기가 입력받은 데이터가 "참"인지 "거짓"인지 구별하는 동안 생성기는 학습 할 때와 똑같은 상태로 가동 되며 테스트 시간에 드롭-아웃을 적용한다는 점에서 일반 적인 프로토콜과는 다르다. 일반적인 연구는 학습된 데이터의 통계가 아닌 시험용 데 이터의 통계를 사용하여 일괄 정규화[30]를 적용하였고, 한 번에 처리하는 데이터의 크기가 1로 설정된 경우, 규격화에 대한 접근방식은 "인스턴스 정규화"라고 하며 이 미지 생성에 매우 효과적이다[31].





제2절 패치로 설계된 판별기

기존의 L2손실과 L1손실은 영상복원 문제에서 모호한 결과를 생성한다는 것은 매우 잘 알려져 있는 사실이다. 이러한 손실의 경우 높은 주파수에서 영상의 선명도를 높이 지는 못하지만 저주파의 경우 매우 선명하다. 판별기의 경우 고주파 영상만을 모델링 하게 되면 판별하기가 훨씬 쉽기 때문에 고주파 영상만을 모델링 하도록 제한하는 동 기를 부여한다.

패치의 규모로 판별기의 구조를 설계하게 되면 판별기의 경우 각각의 $N \times N$ 의 패치 가 영상을 실제인지 거짓인지 분류하려고 한다. 이는 판별기의 최종 출력을 제공하기 위해 영상 전체에 걸쳐 실행되며 모든 응답의 평균값을 판별기에 전달하여 최종출력을 제공한다.

이 방법의 경우 패치의 크기가 영상 전체의 크기보다 훨씬 작을 수 있으며 결과물의 화질도 매우 높다. 패치의 크기가 더 작을수록 작은 매개변수를 사용하기 때문에 실행 속도가 더 빠르며 임의의 큰 이미지에도 적용된다는 이점을 가지고 있다.

패치를 이용한 방법론은 이전[23]에 탐구 되었으며, 이미지의 텍스처 모델과 이미지 의 스타일[24,25,26,27,28,29]의 변형과 전송에서 매우 일반적인 방법이다.

판별기의 최적화를 위해 생성기와 판별기의 기울기 하강을 번갈아가며 한번 씩 수행 해 가며 log D(1 - G(x,z))의 값을 최소화하기 위해 생성기를 학습시키기 보다는 log D(x, G(x,z))의 값을 최대화하는 방향으로 학습을 하면서 목표를 2로 나누면서 판 별기가 생성기에 대해 학습하는 속도를 늦추었다.

판별기에 대한 신경망의 최적화과정에서 원래는 학습데이터 전체에 대한 계산을 수 행해야하지만 너무 많은 계산양이 필요하기 때문에 학습 데이터들의 표본만 추출 후 평균값으로 기울기를 계산했다. 이 방법은 전체 데이터에 대한 계산보다 부정확 할 수





도 있으나 계산속도가 훨씬 빠르기 때문에 같은 시간을 사용하여 더 많은 양의 데이터 를 학습 할 수 있다.

Adam Solver[32]도 같은 목적으로 적용하였는데 이는 목적함수의 최솟값을 찾는 알 고리즘으로 학습율을 0.0002, 학습감소율 각각 0.5, 0.999로 설정하여 진행하였다.

그림 3-4는 판별기를 학습시키기 위해, 생성기를 통해 출력 이미지를 생성하고 판별 기는 입력-목표 쌍과 입력-출력 쌍을 보고 얼마나 현실적으로 보이는지 추측을 한다. 추측 과정이 끝나게 되면 다음 판별기의 가중치는 입력-출력 쌍과 입력-목표 쌍의 분 류 오류를 기반으로 조정된다.



그림 3-4 판별기의 학습과정 영상들을 비교하여 판별기의 가중치 조절함







제4장 실험 결과 및 분석

제1절 실험 환경

학습 및 시험을 위한 시스템 환경은 표4-1과 같이 구성한다. CGAN의 학습을 위한 많 은 물방울 왜곡 영상이 필요했지만 실제 영상이 없어, 영상의 생성이 필요했다. 학습 용 데이터를 준비하기 위해 본 연구에서는 알파혼합을 이용한다.

표	4-1	학습	및	테스트를	위한	환경
---	-----	----	---	------	----	----

CPU	AMD Ryzen 7 1700X Eight-Core Processor 3.40GHz
GPU	GeForce GTX1080
RAM	16.0GB
OS	Microsoft Window10

알파 혼합은 반투명 영상과 배경영상을 결합하여 새로운 혼합 영상을 만드는 과정이 다. 반투명 영상의 투명도는 완전 투명에서 완전 불투명까지 다양하다. 반투명 영상의 색상이 완전히 투명한 경우 혼합 된 색상이 배경영상의 색상이 되고, 반대로 완전히 불투명 한 경우 혼합 된 색상이 반투명영상의 색이 된다. 수식(1)에서 영상의 알파 채 널은 영상에 투명도를 0과 1사이의 값으로 입력한다.

본 연구에서는 투명도를 0.5로 조절하고, 객체사진은 ImageNet[34]에서 제공하는 영 상들을 이용하여 총 2500장을 Google에서 비오는 영상들을 찾아 물방울 왜곡의 특징별 로 정리하여 25장을 이용했다.

각 픽셀마다 알파 값을 곱하기 위해 영상의 크기를 맞추고 판별기가 마스크의 특징

- 17 -





을 학습하지 못하도록 ImageNet에서 준비한 영상들은 순차적으로 Google에서 찾은 영 상들은 임의로 출력하여 합성 하였다.

그림 4-1와 같이 알파 혼합을 사용하여 생성한 학습 또는 시험용 영상에 문제가 발 생하는데, 물방울 왜곡사진의 잔상이 남아 객체의 영상에 흔적이 남는다. 필터를 이용 한 복원영상에도 이러한 잔상이 남아있어, GAN을 이용해 생성된 영상들도 학습을 할 때 잔상까지 학습을 해버려 생성된 영상이 깔끔하지 못하다. 그래서 잔상이 남은 영상 들을 제외하고 다시 학습을 시켜 복원과 생성을 실시했다.



Collection @ chosun





 원영상
 물방울 왜곡영상
 혼합 된 영상

 그림 4-1 학습 및 시험을 위해 혼합에 사용된 영상 및 잔상이 남은 영상

비교평가를 위해 cycle GAN[33]과 중간값 필터, 양방향 필터를 이용해 복원을 진행 하였다.

중간값 필터는 주변 픽셀들과 중심이 되는 픽셀의 값들을 크기에 따라 정렬하고, 중 간값을 선택해서 자신의 픽셀 값으로 하는 필터로 함수의 구성 파라미터는 각각 원본 영상, 출력 영상, 커널의 크기인 3가지로 구성되며, 본 실험에서는 주변의 선이나 모 서리부분이 뭉개져 버리는 단점이 있기 때문에 커널의 크기를 5로 조절하였다.

양방향 필터는 중심이 되는 픽셀에서 거리에 따른 가중치를 적용하여 영상을 부드럽



게 만들어 주는 필터로 중심 픽셀에서의 거리뿐만 아니라 밝기 차이도 고려한다는 점 이 다르고, 함수의 구성은 입력영상, 출력영상, 필터링에 이용하는 픽셀의 지름, 픽셀 이 적용될 색 공간과 거리공간으로 구성되는데 색 공간과 거리 공간의 크기가 커지게 되면 중간값 필터와 마찬가지로 영상이 뭉개져 버리고 코딩 간 오류가 발생하기 때문 에 큰 값을 사용할 수 없어 본 실험에서는 픽셀의 지름을 9로 설정하고 색 공간과 거 리 공간의 크기는 각각 75로 진행하였다.

그림 4-2는 중간값 필터의 커널의 크기를 7로 양방향필터의 각 공간 크기를 100으로 설정하여 복원한 영상으로 원래의 입력영상과도 비교하였을 때 영상이 뭉개져 있음을 확인한다.



력 중간값 필터 양방향 필터 그림 4-2 왜곡 영상과 필터로 복원한 영상의 잘못된 예

공개된 코드를 이용해 CGAN과 cycle GAN의 학습 및 테스트에 사용한다.[15,33] 학습 은 총 400회를 진행 하고, 100회단위로 끊어 학습 데이터 내에 없는 테스트 영상을 150여장씩 생성하며 5가지의 분류 기준으로 나누어 PSNR값의 평균을 계산한다.

PSNR은 최대 신호 대 잡음비로 신호가 가질 수 있는 최대 전력에 대한 잡음의 전력 을 나타낸 것이다. 주로 영상 또는 동영상 손실 압축에서 화질 손실 정보를 평가할 때 사용된다. 신호의 전력에 대한 고려 없이 MSE(평균 제곱 오차)를 이용해서 계산 할 수





있다.

PSNR을 구하는 수식은 다음과 같다.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^{2_I}}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$
(5)

여기서 *MAX*_I는 영상의 최댓값으로, 해당 채널의 최댓값에서 최솟값을 빼서 구할 수 있다. 예를 들어 8bit 그레이스케일 영상의 경우는 255(255-0)이 된다. 단위는 로그 스케일에서 계산하기 때문에 db이다. PSNR 값으로 영상평가는 30db이상이면 영상 간 차이가 크지 않고, 두 이미지의 차이인 에러율도 매우 낮다[33].





제2절. 결과 및 분석

생성된 영상들은 물방울 왜곡의 종류에 따라 5가지로 분류하였다. 생성 영상에 대한 평가는 GAN의 결과물 자체가 새로운 예제이므로 확인할 수 있는 정량적인 척도가 없어 본 연구에서는 비교를 위해 PSNR값을 구해 비교평가를 실시하였다.



영상

중간값 필터

필터 oF



CGAN

영상 H

그림 4-3 일반적인 물방울 왜곡에 대한 복원 영상

첫 번째 왜곡은 그림 4-3와 같이 일반적인 물방울 왜곡 영상을 복원 및 생성하였고, 필터로 복원한 영상의 경우 중간값 필터로 복원된 영상은 양방향 필터보다 물방울의 자국이 많이 지워졌음을 확인하였고,cycle GAN 및 CGAN으로 생성된 영상의 경우 목표 영상과 매우 흡사하다.









cycle GAN

CGAN

목표 영상

그림 4-4 물줄기 및 물안개를 포함한 왜곡 영상에 대한 복원 영상

두 번째 왜곡은 그림 4-4처럼 물줄기 및 물안개가 포함된 왜곡 영상에 대한 복구 및 생성된 영상들이고, CGAN을 이용해 생성된 영상에도 물줄기에 대한 복원이 깔끔하지 못하지만, 필터를 이용한 영상들과 비교해보면 상당히 우수하나 cycle GAN의 생성영상 이 더 결과가 우수하게 나왔음을 확인한다.











그림 4-5 물방울 왜곡이 심한 영상에 대한 복원 영상

세 번째는 그림 4-5과 같은 왜곡으로 이는 다량의 물이 렌즈에 묻을 경우 발생하는 데, 왜곡의 정도가 심해 각 필터를 이용해 복원한 영상은 화질이 입력영상과 비슷해 보이고, 그림 4-5 같이 잔상이 거의 제거 되지 않았지만, cycle GAN과 CGAN의 생성영 상에서 뒤쪽의 가로등까지 복원은 못하였으나 결과가 우수하다.









입력 영상

중간값 필터

양방향 필터



cycle GAN

CGAN

목표 영상

그림 4-6 물줄기가 포함된 왜곡 영상에 대한 복원 영상, cycle GAN과 CGAN의 생성 영상과 목표영상

네 번째는 그림 4-6처럼 물줄기가 흘러내린 왜곡 영상이고, 필터로 복원된 영상은 물줄기가 그대로 남아있지만, cycle GAN과 CGAN으로 생성된 영상은 물줄기에 대한 잔 상이 남아있고 목표영상과 비교했을 때, 영상의 중앙에 위치한 새의 모습도 확실하게 생성이 되지 않았음을 확인할 수 있다.







중간값 필터



cycle GAN

CGAN

목표 영상

그림 4-7 짙은 물안개가 포함된 왜곡 영상에 대한 필터 복원 영상, cycle GAN과 CGAN의 생성 영상과 목표영상

마지막으로, 그림 4-7은 짙은 물안개에 대한 복원영상으로, 필터로 복원한 영상의 경우 오히려 왜곡 영상보다 더 흐려져 있지만, cycle GAN과 CGAN으로 생성된 영상의 경우 목표영상과 거의 유사하지만 자세히 살펴보면 섬의 모양이나 물의 색이 다르다.

위의 5가지 왜곡 영상들에 대한 복원 및 생성된 영상의 PSNR값을 정리하면 표 4-2와 같고, 테스트용 모든 영상을 물방울 왜곡의 5가지 종류별로 분류하여 PSNR값의 평균을 내어 그림 4-8로 나타냈다. 그림 4-9는 CGAN의 학습 횟수별 손실률을 나타낸 것이다. 중간값 필터와 양방향 필터는 비슷한 값이 나왔으나 우수하지 못하였고, cycle GAN과 CGAN으로 생성된 영상의 경우 두 종류의 영상 모두 우수한 결과를 얻었으나 CGAN이 보 다 좀 더 우수한 결과를 얻었음을 알 수 있다.





영상	중간값 필터	양방향 필터	cycle GAN	CGAN
	20.0648 dB	20.1979 dB	25.0172dB	27.0208 dB
	16.0481 dB	16.4928 dB	29.8695dB	29.7493 dB
	20.9353 dB	21.0400 dB	29.1736dB	30.4483 dB
	20.7041 dB	20.7351 dB	26.6996dB	27.1998 dB
	19.5741 dB	19.6263 dB	27.4611dB	29.7802 dB

표 4-2 각 필터로 복원된 영상, cycle GAN, CGAN으로 생성된 영상의 PSNR







그림 4-8 분류에 따른 왜곡별 복원영상의 PSNR 평균

CGAN의 학습 횟수에 따른 손실률



4-9 학습 횟수에 따른 CGAN의 손실률







제 5장 결론

본 연구에서는 효율적으로 물방울 왜곡을 제거하기 위해 기존의 연구들과는 다르게 CGAN을 접목하여 새로운 왜곡 제거 방법을 제안한다. 제안한 방법을 통해 여러 왜곡에 따른 영상을 생성하였고, 생성된 영상을 판단하기 위해 같은 영상들을 중간값 필터와 양방향 필터를 이용하여 복원하고, cycle GAN과 CGAN의 학습 횟수에 따른 영상들을 생 성해 비교를 실시고자 PSNR을 이용하여 수치화 시킨 결과 다른 방법들 보다 매우 우수 한 결과를 얻었음을 보였고, 추가적인 학습을 이용하여 다양한 상황에서 적용이 가능 하기 때문에 관련 작업에 효율적이라고 생각한다.





참고문헌

- [1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio, "Generative Adversarial Networks," arXiv:1406.2661 10 Jun 2014.
- [2] Mehdi Mirza, Simon Osindero, "Conditional Generative Adversarial Nets," arXiv:1411.1784 6 Nov 2014.
- [3] Hsien-Chou Liao, De-Yu Wang, Ching-Lin Yang and Jungpil Shin, "Video-based Water Drop Detection and Removal Method for a Moving Vehicle," Information Technology Journal, Vol. 12 No. 4 pp. 569-583, 2013.
- [4] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," ACM Transactions on Graphics, Vol. 2, pp. 217-236, 1983.
- [5] K. Garg and S. K. Nayar, "Detection and Removal of Rain in Videos," Computer Vision and Pattern Recognition, Vol. 1, pp. 528-535, 2004.
- [6] P. Barnum, S. Narasimhan, and T. Kanade, "Spatio-temporal Frequency Analysis for Removing Rain and Snow from Videos," International Workshop on Photometric Analysis for Computer Vision, 2007.
- [7] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 2nd Edition. Prentice Hall, 2002.
- [8] H. Li, X. Zhang, T. Ng, W. Leow, and Y. Qi, "Rain Removal in Video by Combining Temporal and Chromatic Properties," IEEE International Conference on Multimedia and Expo, pp. 461-464, 2006.
- [9] Mehdi S. M. Sajjadi Bernhard Scholkopf Michael Hirsch, "EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis," IEEE International Conference on Computer Vision 30 Jul 2017.
- [10] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang and Russ Webb, "Learning from Simulated and Unsupervised Images through Adversarial Training," IEEE Conference on Computer Vision and Pattern





Recognition 19 Jul 2017.

- [11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv:1511.06434, 19 Nov 2015.
- [12] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," IEEE Conference on Computer Vision and Pattern Recognition, 25 Apr 2016.
- [13] Phillip Isola, Jun-Yan Zhu Tinghui, Zhou Alexei and A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," IEEE Conference on Computer Vision and Pattern Recognition 22 Nov 2017.
- [14] S. loffe and C. Szegedy, "Batch normalization: Accelerating deep network training," arXiv:1502.03167 11 Feb 2015.
- [15] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," IEEE Conference on Computer Vision and Pattern Recognition 21 Nov 2016.
- [16] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," 14th European Conference, Amsterdam, The Netherlands, 11 Oct 2016.
- [17] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," arXiv:1603.08155, 27 Mar 2016.
- [18] Y. Zhou and T. L. Berg, "Learning temporal transformations from time-lapse videos," arXiv:1608.07724, 27 Aug 2016.
- [19] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixellevel domain transfer," arXiv:1603.07442, 28 Nov 2016.
- [20] G. E. Hinton and R. R. Salakhudinov, "Reducing the dimensionality of data with neural networks," Science, Vol. 313, No. 5786, pp.504-507, 2006.
- [21] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," arXiv:1604.04382, 15 Apr 2016.





- [22] A. A. Efros and T. K. Leung, "Texture synthesis by nonparametric sampling," IEEE International Conference on Computer Vision vol. 2, pp 1033-1038. 20 Sept. 1999.
- [23] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks," arXiv:1505.07376,12, 6 Nov 2015.
- [24] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," SIGGRAPH, pp 341-346.Aug 2001.
- [25] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," SIGGRAPH, pp 327-340. 2001.
- [26] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," IEEE Conference on Computer Vision and Pattern Recognition pp.2414-2423 Jun 1, 2016.
- [27] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," arXiv:1601.04589 8 Jan 2016.
- [28] S. loffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on International Conference on Machine Learning 2 Mar 2015.
- [29] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," arXiv:1607.08022, 27 Jul 2016.
- [30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 , 22 Dec 2014.
- [31] Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A. Efros "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" arXiv:1703.10593, 30 Mar 2017.
- [32] H. Inoue, A. Miyazaki, A. Yamamoto and T. Katsura, "A Digital Watermark Based on the Wavelet Transform and its Robustness on Image Compression" IEEE International Conference on Image Processing, 7 Oct.





1998.

- [33] http://jaejunyoo.blogspot.com/2017/02/deep-convolutional-gan-dcgan-1.html.
- [34] http://image-net.org/.

