[최종보고서]

PIA : POBBAMI in the air

- 밤 영상을 낮 영상으로 바꿔주는 야간순찰 드론

POSCO AI & Big Data Academy 8기
B반 1조
공해인 김소희 박다연 성준혁 장하민 이건우

< 목차 >

1. 프로젝트 개요 **1**

1.1 프로젝트 소개 및 목표
1.2 프로젝트 주제 선정배경
1.3 선행연구 동향

2. GAN 모델소개 **3**

2.1 GAN
2.2 pix2pix
2.3 DiscoGAN
2.4 CycleGAN
2.5 TodayGAN
2.6 최종 모델선정

3. 데이터 수집 **10**

3.1 데이터 수집 및 생성
3.2 데이터 전처리
3.3 코딩 구현

4. 모델튜닝 및 결과 **13**

4.1 Hyper Parameter 수정
4.2 CycleGAN의 학습 안정성을 위한 구조 수정
4.3 최종결과

5. 드론 **17**

5.1 드론 선정사유
5.2 드론 환경설정
5.3 드론 장거리 무선통신 환경 구축
5.4 프로젝트 구조도
5.5 프로그래밍 이슈

6. 기대효과 및 보완사항 **29**
6.1 기대효과
6.2 보완사항

7. 팀원소개와 상호평가 **30**
7.1 상호평가

8. 선행연구(학습모델) **33**
6.1 Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
6.2 Night-to-Day Image Translation for Retrieval-based Localization
6.3 On the replication of CycleGAN

9. Reference **37**

1. 프로젝트 개요

1.1 프로젝트 소개 및 목표

본 프로젝트명은 〈PIA: POBBAMI In the Air〉 이다. 본 프로젝트의 목적은 드론으로 수집한 야간 영상을 낮 영상으로 변환하는 것이다.

1.2 프로젝트 주제 선정배경

본 프로젝트의 주제는 기존 CCTV의 한계점을 보완하기 위해 선정되었다. 먼저, 기존 고정형 CCTV는 물리적 한계가 있다. 고정형 CCTV는 한곳에 부착되어 사용되므로 촬영 범위에 한계가 존재하며, 넓은 범위를 촬영하기 위해서는 여러 곳에 CCTV를 설치해야 한다. 또한 CCTV가 부착된 위치나 각도에 따라 사각지대가 발생한다는 문제점이 있다. 이에 더해서, 일반 카메라를 사용하는 CCTV는 야간 시 사람 및 사물에 대한 식별 능력이 떨어진다. 많은 범죄가 야간에 일어난다는 점을 고려하면, 야간 감시 기능이 미흡한 CCTV 화면은 큰 문제라고 볼 수 있다.

드론은 기존 CCTV의 물리적 한계점을 극복할 수 있다. 드론은 여러 장소를 이동하며 촬영할 수 있다는 점에서 넓은 범위를 촬영할 수 있고 사각지대 문제 또한 해소할 수 있다. 따라서 본 프로젝트는 드론이 새로운 CCTV의 형태로 사용될 수 있다는 점에 주목하였다. 또한 야간 영상에서 물체 식별이 잘되지 않는 문제점을 보완하기 위해 이미지 변환 알고리즘을 적용하여 야간 영상을 주간 영상으로 변환하고자 한다.

정리하면, 본 프로젝트는 드론이 공중에서 수집한 야간 영상을 딥러닝을 사용하여 밝은 주간 영상으로 변환해 사용자에게 제공하고자 한다.

1.3 선행연구 동향

가. 선행연구 분석

프로젝트에 들어가기 앞서 이미지 변환을 다룬 선행 연구를 조사해보았다. 안남현과 강석주(2018)는 CycleGAN 모델을 사용하여 야간 도로 영상을 주간 도로 영상으로 변환하는 연구를 수행했다. CycleGAN 모델은 기존의 밝기 변환 알고리즘인 히스토그램 평활화와 감마 보정에 비해서 더 우수한 성능을 보였다. 구체적으로, CycleGAN 은 기존 알고리즘에서 나타나는 열화 현상 없이 밝기를 보정하였으며 단순히 영상 전체의 밝기를 증가시키는 것이 아닌 완전한 주간 영상으로 변환시켰다. 저자들은 이러한 결과를 토대로 CycleGAN 으로 생성한 주간 영상이 기존 방법들에 비해 시각적으로 개선된 영상을 생성하였다고 주장하였다.

조상흠 등(2019)는 CycleGAN 으로 변환한 영상을 학습 데이터로 사용하여 물체 검출(object detection)에 활용할 수 있음을 밝혔다. 저자들은 야간 데이터셋의 부족으로 인해 야간 시의 물체 검출 알고리즘의 성능이 좋지 않음을 지적하며, 이러한 문제를 이미지 변환 알고리즘을 이용해서 해결하고자 하였다. CycleGAN을 사용하여 주간 영상을 야간 영상으로 변환하고, 변환한 영상을 야간 물체 검출을 위한 학습 데이터셋으로 사용하였다. 그 결과, 주간 영상만을 학습 데이터로 했을 때보다 주간 영상과 변환된 야간 영상을 학습 데이터로 사용하였을 때 야간 물체 검출에 있어서 높은 성능을 보였다. 따라서 본 연구는 CycleGAN 으로 생성된 데이터도 학습에 사용될 수 있음을 보였다.

선행 연구조사 결과를 정리하면, 1) 이미지 변환 알고리즘으로 CycleGAN 이 주로 사용되고 2) CycleGAN

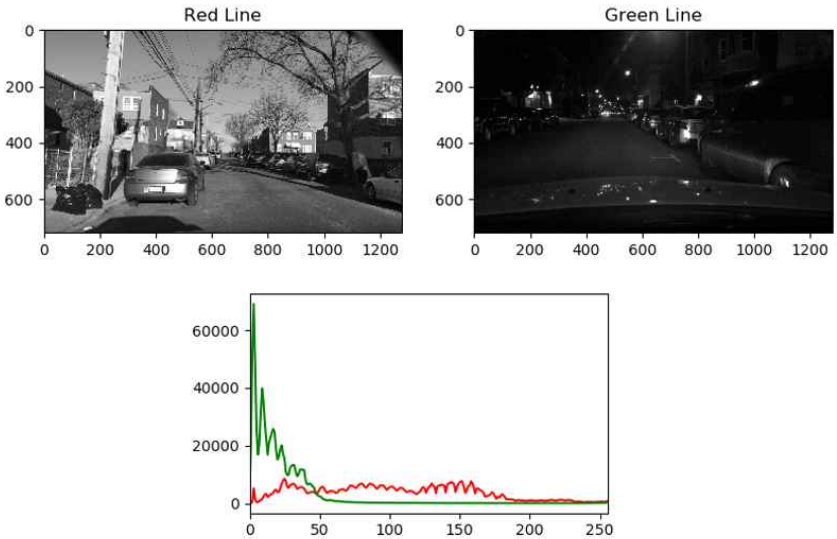
으로 생성한 이미지를 학습 데이터로 사용할 수 있다.

나. 낮, 밤 이미지 기준

관련 선행 연구를 조사한 결과, 낮과 밤 이미지에 대한 구체적인 기준이나 설명을 찾아볼 수 없었다. 낮과 밤 이미지에 대한 기준을 설정하기 위해, 선행연구에서 자주 사용되는 Berkeley Deep Drive 데이터셋 (이하 BDD)을 조사해보았다.

먼저 BDD의 낮, 밤 이미지 픽셀들의 평균값을 조사하였다. 이를 위해 opencv를 사용해 이미지의 히스토그램을 추출했다. 이미지에서 히스토그램은 이미지를 구성하는 픽셀값 분포에 대한 그래프이다. X축은 픽셀값으로 범위는 0 ~ 255 사이이고, Y축은 이미지에서 해당 픽셀값을 가진 픽셀의 개수를 의미한다. 히스토그램의 왼쪽에는 가장 어두운 검은색 픽셀의 갯수를 보여주며 오른쪽으로 갈수록 밝은 픽셀의 갯수를 보여준다.

BDD의 밤, 낮 이미지의 픽셀값 분포를 본 결과, 밤 이미지는 분포가 0에 치우쳐져 있고, 낮 이미지 는 고르게 분포되어 있는 것을 볼 수 있었다.



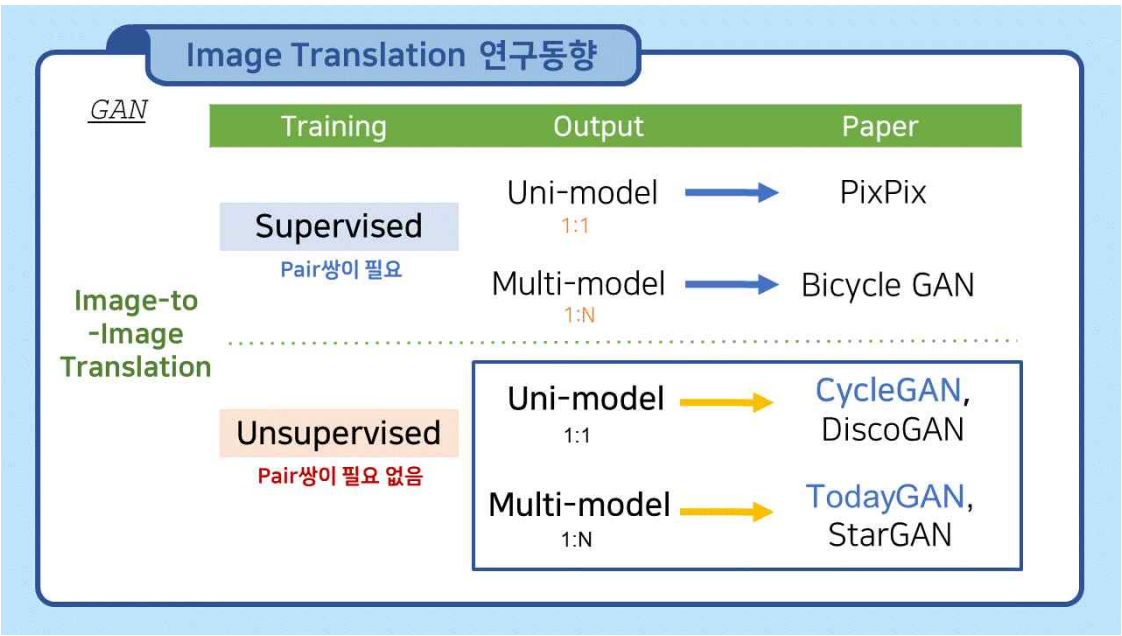
BDD 데이터셋의 밤, 낮 이미지 10개를 추출하여 픽셀값 분포의 평균값을 본 결과, 낮 이미지의 평균 픽셀값 범위는 80-143이었고, 밤 이미지의 평균 픽셀값 범위는 14-64 이었다. 이를 통해 낮 이미지의 경우 평균 픽셀값이 80 이상을, 밤 이미지의 평균 픽셀값이 64 이하를 기준으로 잡을 수 있다. 하지만 BDD 데이터는 도로 주행 환경으로 본 프로젝트에서 사용될 데이터와 성격이 달라, 해당 기준을 참고하는 수준으로만 진행하고자 한다.

드론으로 수집한 낮, 밤 이미지가 위의 기준을 충족시키는 지 알아보기 위해 이미지 10개를 추출해서 평균 픽셀값을 알아보았다. 그 결과, 낮 이미지의 평균 픽셀값은 85 이상이고 밤 이미지의 평균 픽셀값은 10 이하였다. 이는 위의 기준을 충족하기 때문에 해당 이미지를 학습 데이터로 사용하는데 문제가 없다고 판단하였다.

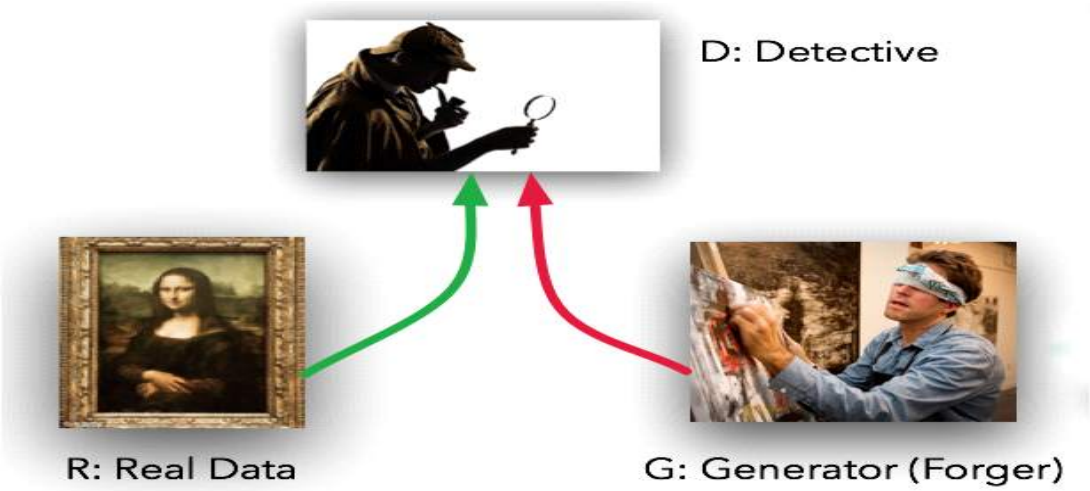
2. GAN 모델 소개

1. GAN

가. GAN이란 무엇인가?



GAN은 생성적 적대 신경망(generative adversarial networks)으로 불리며, 비지도 방식 러닝머신을 사용해 데이터를 생성한다. GAN은 생성기 신경망(Generator network)과 판별기 신경망(discriminator network)이라고 하는 두 가지 신경망으로 구성된 심층 신경망 아키텍쳐이다. 두 신경망은 서로 의존하면서 여러 세대에 걸쳐 생성하고 판별하기를 반복하는 식으로 서로 대항해서 훈련하게 한다.



생성기 신경망(Generator network)는 기존 데이터를 사용해 신규 데이터를 생성해낸다. 예를 들어, 기존 그림을 사용해 새로운 그림을 생성해 내는 식이다. 반면에, 판별기 신경망(discriminator network)은 진짜 데이터와 생성기 신경망이 생성해 낸 데이터 중에 어떤 게 진짜인지 구별해 내려고 애를 쓴다. 일반적으로 이진 분류를 한다. 위의 그림을 보면 이해가 쉽다. Generator는 최대한 X와 유사한 그림을 만들어 내려고 하고, Discriminator는 그것을 최대한 올바르게 판정하려고 하는 경쟁구도라고 말할 수 있다. 아래의 그림은 전체적인 GAN의 구현 방식이다.

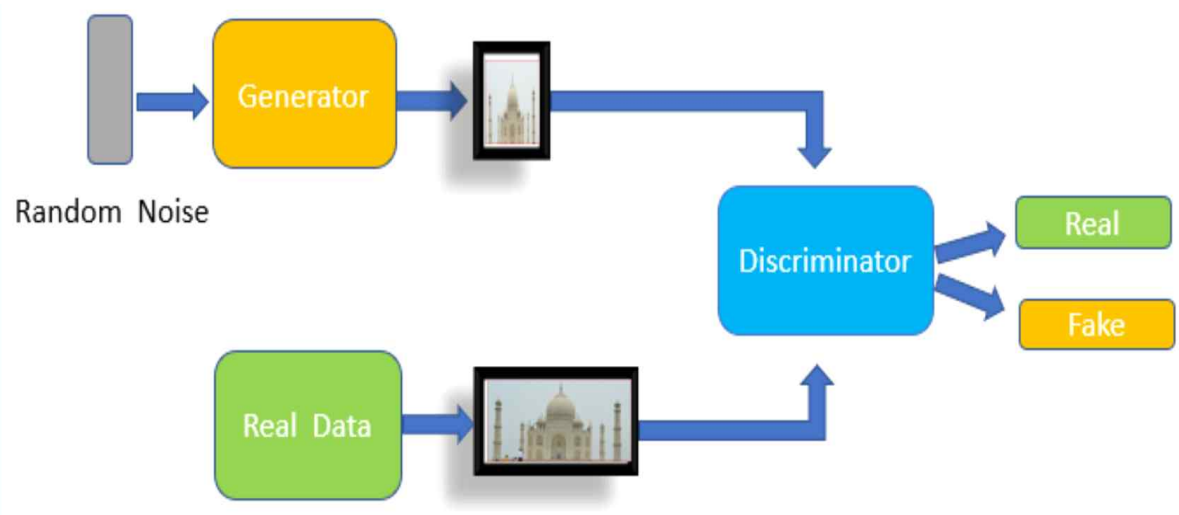
나. GAN의 이점

a. GAN은 비지도 학습 방식이다. GAN은 분류된 데이터가 필요하지 않고 레이블이 없는 데이터를 사용하여 학습하므로 데이터 처리 과정의 시간을 줄일 수 있다.

b. GAN은 데이터를 생성한다. GAN은 실제 데이터와 유사한 데이터를 생성한다. 실제로 GAN은 실제 데이터와 구별할 수 없는 이미지, 텍스트, 오디오 및 비디오를 생성할 수 있다.

c. GAN은 데이터의 밀도 분포를 학습한다. GAN은 지저분하고 복잡한 데이터의 분포를 학습할 수 있다.

d. 훈련된 판별기는 일종의 분류기다. 훈련을 마치고 나면 우리는 판별기와 생성기를 얻게 된다. 판별기 신경망은 일종의 분류기여서 물체들을 분류하는 데 사용할 수 있다.



다. 실용적인 GAN 애플리케이션

1. 이미지 생성: 생성적 신경망(generative networks)은 표본 이미지를 가지고 훈련을 받은 뒤에 진짜 같은 이미지를 생성해 낼 수 있게 된다. 훈련을 마치고 나면은 훈련 집합에 들어 있는 이미지들과는 다른 새로운 이미지들을 생성해 낼 수 있게 된다. 이러한 이미지 생성 기능을 사용하여 마케팅 업무, 엔터테인먼트 산업, 로고 작성, 소셜 미디어 분야 등에서 사용될 수 있다.

2. 텍스트 대 이미지 합성: 서술된 문장을 바탕으로 이미지를 생성해 낼 수 있다. 이런 경우, 영화산업과 만화 사업(이야기 줄거리 자동 생성) 등에서 사용될 수 있다.

3. 얼굴 노화: 이 기능은 엔터테인먼트 산업과 감시 산업에 유용하다. 사람들이 나이가 들어도 얼굴 인식을 바탕으로 얼굴 검증을 하는 보안 시스템을 바꾸지 않아도 되고 이미지를 여러 연령대에 맞게 생성할 수 있으므로 강력한 얼굴 검증용 모델을 훈련하는 데 사용 가능하다.

4. 이미지 대 이미지 변환: 이미지 대 이미지 변환은 낮 화면을 밤 화면으로 바꾼다거나, 사진을 피카소나 반 고흐의 작품처럼 보이도록 화풍을 바꾸는 등의 이미지 변환이 가능하다. 이러한 기능을 이용하여 시간을 절약할 수 있다.

5. 비디오 합성: GAN을 비디오(동영상)을 생성해내는 데도 사용할 수 있다. 짧은 시간의 비디오 생성을 가능하게 하고 영화 제작자의 생산성을 높이게 할 수 있다.

6. 고해상도 이미지 생성: 저해상도 카메라에서 찍은 사진이 있는 경우라면, GAN은 필수 세부 정보를 잃지 않으면서도 고해상도 이미지를 생성할 수 있도록 한다. 웹사이트에 유용하다.

7. 이미지 결함 보정: 결함이 있는 이미지 같은 경우, GAN으로 해당 부분들을 복원할 수 있다.

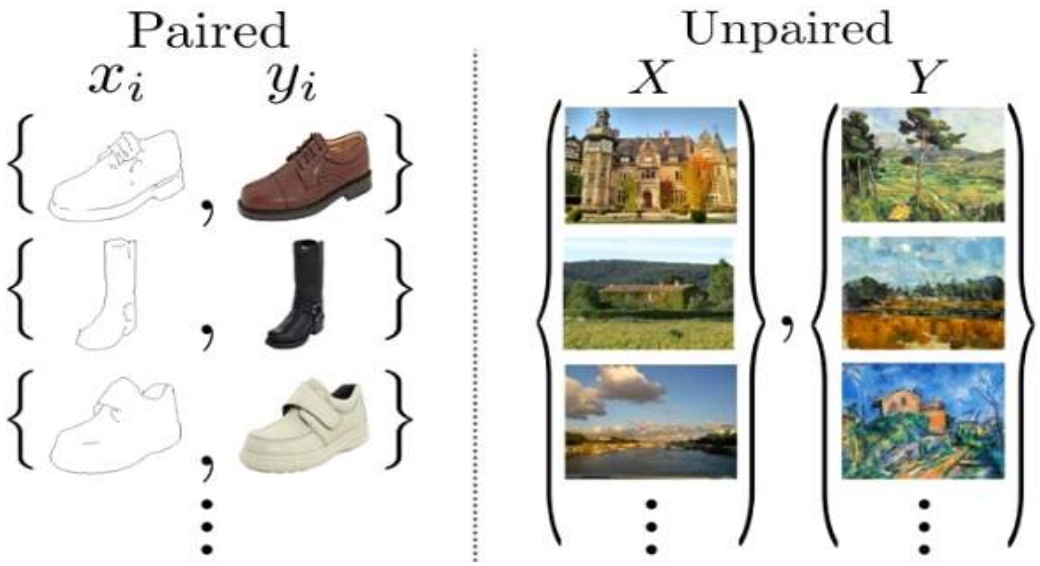
라. GAN의 종류

GAN의 종류로는 수천 가지가 되며 대표적인 일곱개의 아키텍쳐로 DCGAN, StackGAN, CycleGAN, 3D-GAN, Age-cGAN, pix2pix 가 있다. 우리조는 총 3개 pix2pix, DiscoGAN, CycleGAN의 모델을 사용후보로 결정하였다.

2. pix2pix

pix2pix는 GAN 종류중 하나로서 이미지 대 이미지 변환에 사용된다. 다른말로 하면은 이미지를 한 가지 표현에서 다른 표현으로 변환하는데 pix2pix를 사용한다. pix2pix는 입력 이미지를 출력 이미지로 사상(mapping)하는 방법을 학습한다. pix2pix는 진짜와 유사하고 깔끔한 이미지를 생성하는 잠재력이 크다.

pix2pix는 여러 GAN과 동일하게 생성기 신경망과 판별기 신경망이라는 두 가지 신경망으로 이루어진다. 생성기 신경망의 아키텍처는 U-NET의 아키텍처에 영감을 받은 것이다. 또한, 판별기의 신경망의 아키텍처는 PatchGAN의 아키텍처에서 영감을 받은 것이다. pix2pix와 다른 GAN들의 가장 큰 차이점은 pix2pix는 paired 데이터가 필요하다는 것이다. 아래의 그림에서 볼 수 있듯이 pix2pix는 paired 데이터를 필요로 하므로, 모델을 학습을 시키려면 x1과 y1이 있어야 한다. 하단의 신발 예시에서 x1의 이미지는 신발의 edge의 그림이고 y1의 데이터는 신발 전체의 이미지이다. 또한, 다른 데이터들도 동일한 방식으로 나열되어 있다. 결과적으로 pix2pix는 paired 데이터를 필요로 한다는 한계가 존재한다.



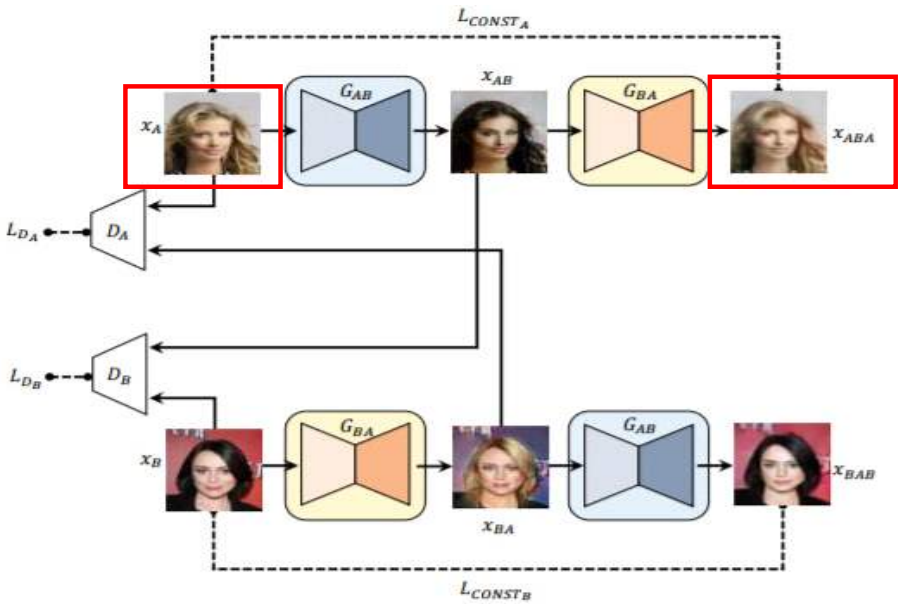
반면, DIsco GAN과 CycleGan의 경우 unpaired 데이터를 사용하기 때문에 pix2pix보다는 제약이 적다. 예를 들어, 최근의 사진을 반고흐의 그림으로 변환하고자 할 때, CycleGAN 모델로는 구현 가능하지만 pix2pix로는 불가능하다. 그 이유는 cycleGAN에서는 X에 사진, Y에 반고흐 그림을 넣고 학습 시키는 반면, pix2pix는 동일한 사진을 반고흐가 그리지 않는 이상 paired 데이터 수집 자체가 어렵기 때문이다. 따라서 pix2pix는 다른 GAN 모델들보다 제약이 많다, 이어서 GAN의 대표인 DiscoGAN과 CycleGAN을 살펴보자.

3. DiscoGAN

DiscoGAN은 discovery-GAN의 줄임말이다. DIscoGAN의 가장 큰 특징은 cross-domain을 GAN에 적용한 것이다. 하단의 그림에서 볼 수 있듯이, input에 특정 스타일의 가방 이미지를 넣으면 output으로 유사한 스타일의 신발을 만들어 내는 모델이 DiscoGAN이다.

(b) Handbag images (input) & **Generated** shoe images (output)

DiscoGAN은 도메인을 변경하는 특정 함수를 찾아낸다는 의미에서 Discovery-GAN으로 불리는데, 아래 그림은 DiscoGAN의 아키텍쳐를 나타내고 있다. reconstruction 이미지(X aba)와 실제 이미지(Xa)의 차이를 줄이는 방식으로 학습한다. DiscoGAN은 학습 속도는 빠르며 이미지의 형태 변화가 큰 데이터에 적합하다.

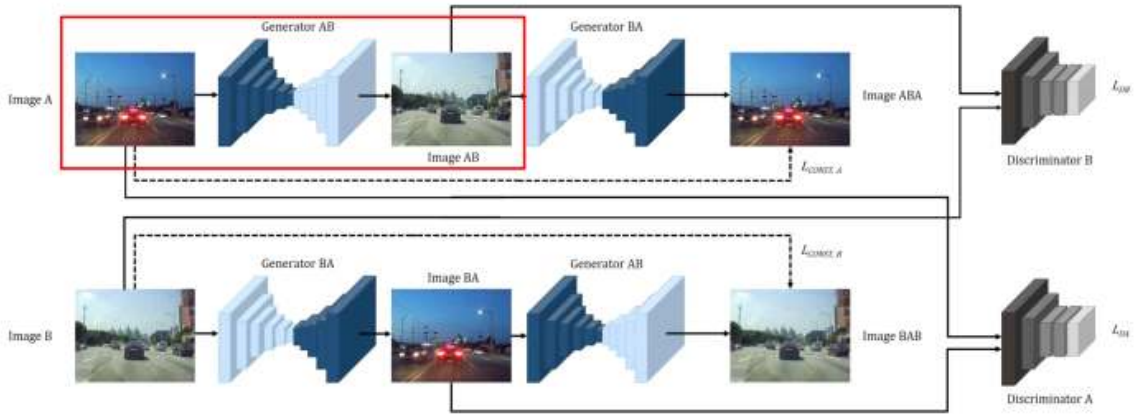


4. CycleGAN

CycleGAN은 이미지의 화풍(style)을 변경하거나, 그림을 사진으로 바꾸거나 사진을 사진으로 바꾸는 등 같은 domain간의 transfer 작업을 기반으로 한 GAN 모델이다. 화풍 모사의 많은 애플리케이션에서 짝 데이터(pair)는 훈련에 필수적인 부분이다. CycleGAN은 낡아빠진 틀을 깨고 나와 짝 입력이 없는 모델을 안정적으로 훈련할 수 있었던 최초의 GAN 구현 중 하나이다.

CycleGAN에서는 생성기와 판별기로 아주 간단한 합성공 신경망(CNN, Convolutional neural networks)을 사용한다. 아키텍쳐는 A에서 B로, B에서 A로 향하는 표현(representation)을 학습할 수 있게 훈련하기 위해 이 신경망들을 봉합(stich)한 방식으로 이해하면 된다.

CycleGAN은 생성기 신경망을 학습하려고 시도한 뒤에, 번갈아가며 두 가지 mapping을 학습한다. CycleGAN은 대부분의 GAN 모델에서 단일 생성기 생성기 신경망을 교육하는 대신 두 개의 생성기 및 두 개의 판별기 신경망을 교육한다. 아래의 그림과 같이 생성기가 두 개(Generator AB와 Generator BA)가 있고, 판별기 두 개(Discriminator A와 Discriminator B)가 있다.



CycleGAN은 두 개의 데이터가 각각의 생성기를 통해서 generated image로 만들어지며 generated image를 다른 생성기를 통하여 reconstructed image로 변환시켜, 처음에 input image와 reconstructed image의 차이를 최대한 줄이는 Cycle형식의 방식으로 CycleGAN이라는 이름이 붙여졌다. 예를 들어, 위의 그림에서 ImageA가 generatorAB를 통한 ImageAB라는 generated image가 생성된다. 그리고, image AB는 DiscriminatorB를 통하여 얼마나 ImageB와 유사한지 판별하고, generatorBA를 통하여 다시 ImageABA라는 reconstructed image로 생성된다. 그리고 마지막으로 imageA와 imge ABA의 유사도, 즉 얼마나 차이가 나는지의 Loss를 통하여 학습하고 싸이클 형식으로 반복된다. 아래의 cycle-consistency loss를 최소화하는 방식으로 구현이 된다. CycleGAN은 학습 속도는 느리지만 원래의 형태를 유지한 채 스타일을 변환할 때 적합하다.

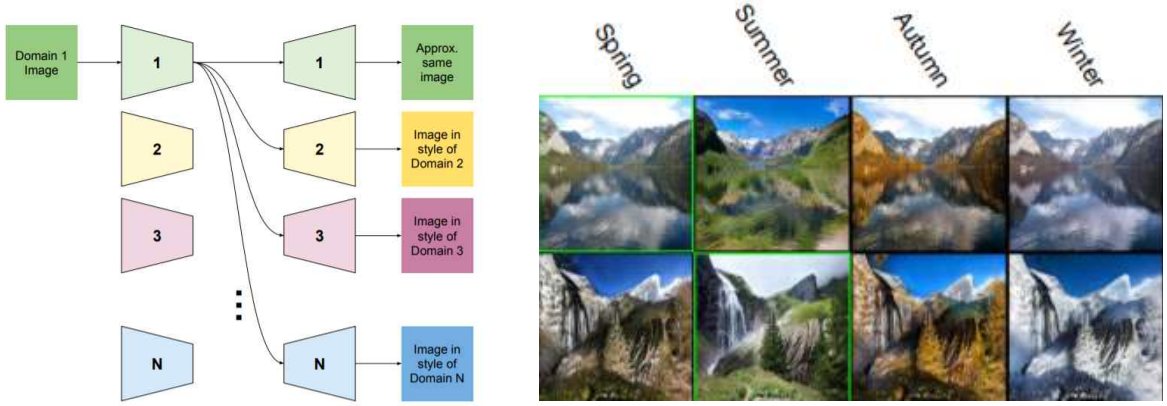
Cycle-consistency Loss

$$L_{cyc}=E_{x\sim p_{data}(x)}[\|F(G(x))-x\|_1]+E_{y\sim p_{data}(y)}[\|G(F(y))-y\|_1]$$

5. TodayGAN

TodayGAN은 CycleGAN을 Multi-domain에 사용할 수 있도록 확장한 ComboGAN의 발전된 모델으로, 세부 네트워크를 변경하여 여러 domain의 변형을 가능하게 하면서 이미지의 퀄리티도 상승시켰다.

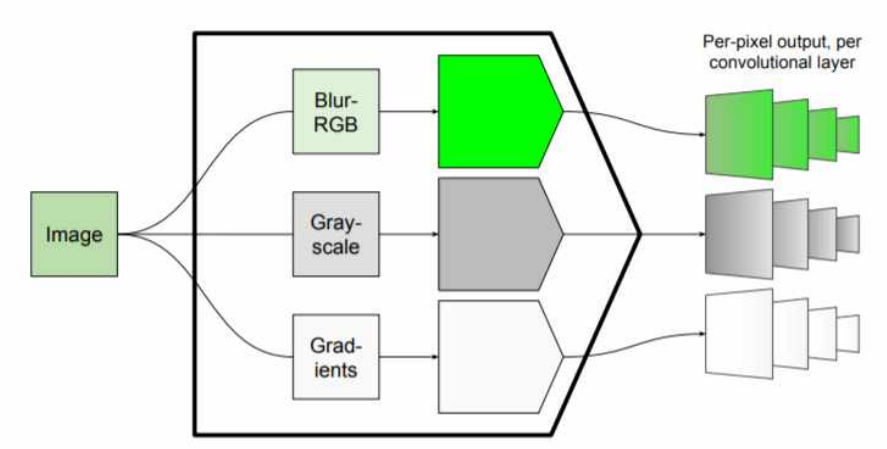
전체적인 구조는 CycleGAN과 동일하게 Image를 X->Y 변환 후 다시 Y->X'변환시켰을 때 X와 X'의 차이를 뜻하는 Cycle-consistency loss 최소화하는 형태이며 구체적인 구조는 Generator의 경우, Multi-domain변형을 위해 여러 개가 묶여있는 pair의 형태를 가진다는 것을 빼고는 차이점이 없다. 하지만 본 프로젝트에서는 밤-낮 두 개의 domain만 사용하므로 Generator는 CycleGAN과 동일하다고 할 수 있다.



(좌:Generator구조,우:Multi-domain변환된 이미지)

큰 차이점이 발생하는 곳은 Discriminator인데, 먼저 TodayGAN은 이미지의 화질개선을 위한 모델인 WESPE의 아이디어에서 착안하여 각 Discriminator가 3개의 필터를 가지는 구조로 변경하였다. 각 필터는 Input의 RGB,Grayscale(명도),Gradients(엣지검출을 위한 변화도)를 처리하여 이미지의 Texture, Color, Gradients를 잘 표현하며 이미지를 생성한다.

$$\mathcal{L}_{GAN}(G_A, D_B, A, B) = \begin{cases} \mathbb{E}_a\mathbb{E}_b[(D_B(b) - D_B(G_A(a)) - 1)^2] \\ \quad \textit{from Discriminator perspective} \\ \mathbb{E}_a\mathbb{E}_b[(D_B(G_A(a)) - D_B(b) - 1)^2] \\ \quad \textit{from Generator perspective} \end{cases}$$



6. 최종 모델선정

최종적으로 동 프로젝트에서는 GAN이라는 deep learning을 기반으로 night to day image translation이 적합한 GAN 모델들을 비교해보았다. 영상 변환에 적합한 pix2pix, DiscoGAN, CycleGAN 모델의 장단점을 분석한 결과, 동 프로젝트에서는 드론을 통해 수집한 데이터를 바탕으로 night to day image translation을 진행함에 따라 paired 데이터를 구하는 것이 불가능하다고 판단하여 pix2pix 모델을 제외하였다.

또한, 학습 속도가 느리더라도 형태가 아닌 스타일의 변화를 해야하므로 DIscoGAN 보다는 CycleGAN이 적합하다고 판단하여 CycleGAN을 이용하여 드론을 이용한 night to day image translation의 프로젝트를 진행하였다. 하지만, CycleGAN의 학습 과정 중 일정 결과 이상의 모델 구현이 어려웠으며, 파라미터 조절을 통해 CycleGAN 모델의 결과를 높이는 동시에 2019년 최신 GAN모델 중 하나인 TodayGAN을 통해 동시에 두 모델의 학습을 진행하기로 결정하였다.

모델	Pix2Pix	DiscoGAN	CycleGAN	TodayGAN
Data set	대응되는 한 쌍의 데이터(paired)	짝이 없는 데이터(unpaired)		
특징	학습속도 느리며 큰 변화를 주기 어려움	학습속도가 빠르며 형태에 변화가 큰 데이터에 적합	학습속도 느리고 원래 형태 유지한 채 스타일 변환할 때 적합	CycleGAN의 성능을 개선한 모델로 밤낮 이미지 변화에 특화
구조	1개의 생성자, 1개의 판별자로 이루어진 2개의 convolution layer를 사용하여 이미지를 얻는 방식	2개의 생성자, 2개의 판별자로 이루어진 4개의 convolution layer를 사용해 각 도메인의 이미지를 처리하여 이미지를 얻는 방식		2개의 생성자, 2개의 판별자로 이루어지며, 각 판별자가 3개의 필터(RGB, 명도, 엣지검출)로 구성되어 이미지를 생성
결과	기각		채택	채택

3. 데이터 수집

동 프로젝트의 목적은 일몰 이후 야간 촬영 이미지를 낮 이미지로 변환하는 것이므로, 제대로 된 결과물 도출을 위해 기존 데이터의 활용 및 신규 데이터 수집 등 다양한 방안을 고려하였다. 참고한 기존 데이터셋의 경우 자율주행 차량의 시야에서 촬영된 영상으로, 드론에 학습시켜 적용하는 것은 무리가 있다고 판단하여 bebop 드론을 통해 직접 낮, 밤 영상을 촬영하여 학습 데이터를 생성하였다.

3.1 데이터 수집 및 생성

가. 기존 데이터 분석

야간 이미지 변환을 활용한 선행연구를 탐색하던 중 KAIST에서 자율주행 차량으로 야간 및 주간 운전 시 보행자를 detection 하는 연구를 확인하였다. 해당 dataset은 campus, load, Downtown으로 구분되어 있으며, Day이미지 (train) 33,399장/ (test) 29,719장, Night 이미지 (train) 16,788장/ (test) 15,962장으로 구성되어 있다.

해당 이미지를 바탕으로 학습시킨 결과 제대로 된 학습이 이루어지지 않았으며, 학습 이미지의 시야 각도와 드론 주행 시 시야 각도의 차이가 크다고 판단하여 직접 데이터셋을 수집하는 방향을 고려하였다.

나. 데이터 수집

[1차 데이터 수집]

- DAY : 오후 12시, NIGHT : 오후 4시 30분 ~ 5시 15분
- 구역 : 연구 4동 앞 광장, 본관으로 이어지는 대로변

데이터 수집에 앞서 선행연구를 분석한 결과 이미지 수집에 대한 명확한 기준에 관련된 연구는 찾아보기 어려웠다. 이에 시간대별로 데이터를 수집하여 비교해보는 방법을 고려하였는데, 낮 data는 오전 9시부터 일몰 이전까지의 오후 시간대에 수집하는 것이 가장 적절하다고 판단하여 데이터를 수집하였다. 밤 data는 11월 말 일몰시간이 평균 오후 5시~ 5시 10분 전후인 것으로 확인됨에 따라, 일몰시간 15분 이전부터 일몰 시간대까지 수집하여 비교하였다. 일몰시간 이후에도 데이터 수집을 진행하였으나, 비교결과 일몰 시간 이전에 촬영한 이미지와 큰 차이가 없어 일몰 15분 이후 데이터를 모델 학습 및 테스트에 활용하였다.

[2차 데이터 수집]

- DAY : 오후 4시 ~ 4시 반, NIGHT : 오후 7시 30분 ~ 8시
- 구역 : 연구 4동에서 RIST 식당으로 이어지는 대로변 및 뒷길

1차 데이터 수집을 통해 학습을 진행하였으나 초기 기대한 결과를 얻기 힘들었다. 참고 문헌에서 사용한 밤 데이터의 경우에도 사물의 분간이 가능한 초저녁 정도의 이미지를 활용하였기에, Gamma 조정, histogram equalized 등을 통해 이미지 전처리 후 학습시키는 방법을 시도했으나 오히려 해상도가 낮아져 결과는 크게 다르지 않았다. 이러한 원인은 학습모델이 제대로 구축되지 않았거나, 학습 데이터셋이 적합하지 않은 것으로 결론짓고 다른 데이터 셋을 모델에 학습시켜 이를 검증하고자 하였다. 검증결과 다른 데이터셋의 경우 기대하는 수준의 결과를 보임에 따라 2차 데이터 수집을 진행하였다.

Night 이미지의 경우 사물이 분간이 가능한 구역에서 수집하는 것이 필요하므로, 연구 동 근처에서 비교적 밝은 구간을 탐색하여 연구 4동에서 RIST 식당으로 이어지는 대로변과 뒷길에서 데이터 수집을 진행하였다. 해당 구간은 사람, 차량의 이동 및 주변의 건물이 많지 않고, 가로등이 다수 존재하고 있어학습 시 불필요한 물체가 생성될 가능성을 낮출 것이라 판단하였다. 낮과 밤 모두 동일한 구간에서 낮 오후 4시, 밤 오후 7시 30분 ~ 8시까지 데이터를 수집하여 모델 학습 및 테스트에 활용하였다.

3.2 데이터 전처리

드론에서 촬영된 영상을 모델에 학습시키기 위해서는 영상을 프레임으로 변환하는 전처리가 필요한데, 이번 프로젝트에서는 각각의 낮 영상과 밤 영상을 20 frame 단위로 추출하였다. 한 개의 영상이 3분정도 되는 경우 추출되는 이미지는 400개 정도로 png파일 형태로 저장하였다. CycleGAN 모델의 경우 이미지 사이즈가 일정 사이즈 이상인 경우 학습이 불가능하므로 이미지 사이즈의 조절을 진행하였으며, TodayGAN 모델의 경우 별도의 데이터 전처리가 불필요한 상황이다.

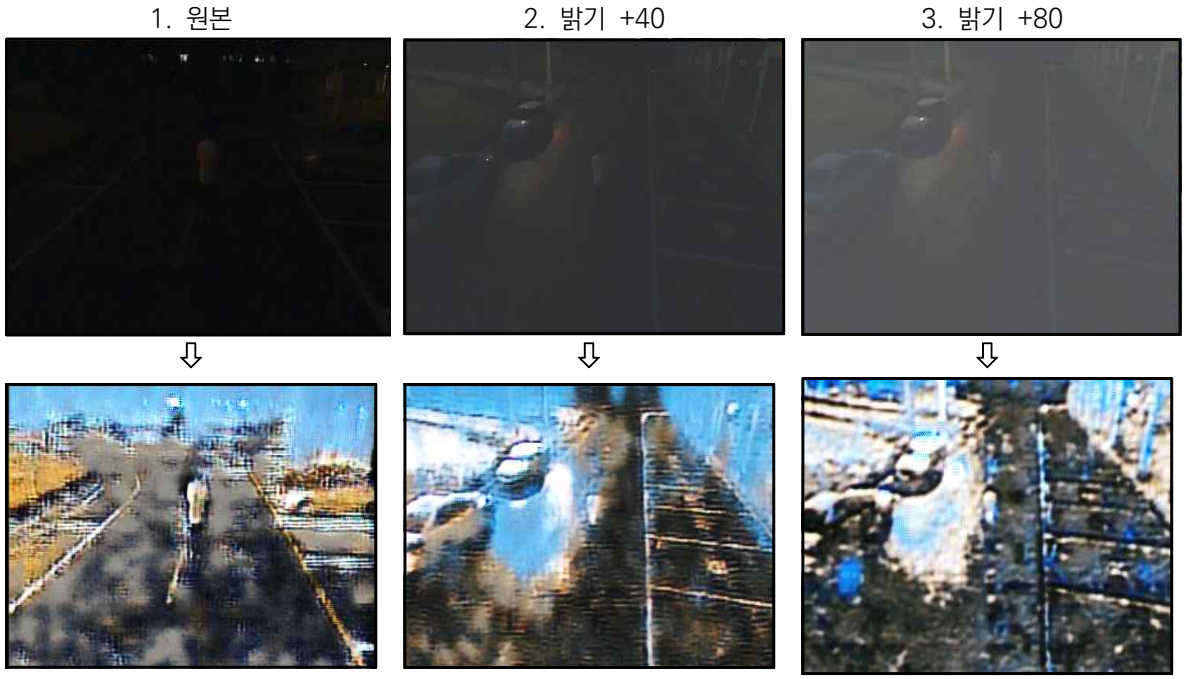
가. 이미지 사이즈 조절

영상을 frame으로 변환 시 Image shape이 바뀌기 때문에 이미지 추출 후 일괄적으로 이미지 Resize를 진행하였다. 기존 이미지 사이즈 1318*741에서 불필요한 부분을 잘라내고 CycleGAN 학습 시 640*360 사이즈 이미지로 조절하였다.

나. 밝기 조절

낮 이미지의 경우 별도의 전처리 없이 학습 데이터를 사용하였으며, 밤 이미지의 경우 밝기 조정(원본, 밝기 + 40, 밝기 +80)을 통해 학습을 진행하였다. 각각의 데이터로 학습한 결과, 밝기 조절을 하지 않은 원본의 변환 결과가 가장 좋은 것으로 분석되어 별도의 밝기 조절은 적용하지 않았다.

(※참고) 낮과 밤의 구분 기준에 대한 설명은 선행연구조사 2) 낮, 밤 이미지 분석내용 참고(보고서 p.3)



다. Gamma 조절

이미지의 전처리 시 Gamma 조절을 통해 밝기를 조정하는 연구논문을 참고하여, 동 모델에서도 Gamma 별 결과값에 차이가 있는지 분석하였다. Gamma의 변화를 통해 밤이미지가 밝아지는 결과를 얻어지만, 같은 Gamma 별 동일한 조건에서 학습진행 시 뚜렷한 결과의 차이는 존재하지 않았다. 따라서 이미지는 리사이즈 외 별도의 전처리는 진행하지 않고, CycleGAN학습을 진행하였다.

1. 원본	2. Gamma +1.5	3. Gamma +2.5
		

3.3 코딩 구현

모델 개발 환경은 ubuntu 16.04, conda 4.7.12, keras 2.3.1, tensorflow 1.15.0에서 구현하였다.

〈 이미지 Crop 코드 〉

```
from PIL import Image
import os.path, sys

path = ('/home/pirl/Desktop/img/Dataset/trainB')
dirs = os.listdir(path)

def crop (count ):
   for item in dirs:
       fullpath = os.path.join(path,item)  #corrected
       if os.path.isfile(fullpath):
           im = Image.open(fullpath)
           f, e = os.path.splitext(fullpath)
           imCrop = im.crop((264, 0 , 1054 , 360 )) #corrected
           # imCrop.save(q + 'Cropped.png', "png", quality=100)


imCrop.save('/home/pirl/Desktop/img/Dataset/trainB/trainBB/frame%d
.png' %count, quality =100 )
           count += 1

crop(0 )
```

4. 모델 튜닝 및 결과

4.1 Hyper parameter 수정

CycleGAN의 경우 기존 default 값으로 학습 시 만족할만한 수준의 결과에 도달하지 못하였다. 따라서 Train, Test image, Image size를 고정한 상태에서 각 Hyper parameter를 조정하고 결과를 비교하는 방식으로 최적의 parameter를 결정하였다. 초기 예상했던 결과와 달리 Epoch수가 증가한다고 해서 모델의 학습결과가 개선되지 않는 모습을 보여 Loss그래프를 비교하여 적절한 epoch수를 결정하였다.

TodayGAN의 경우 CycleGAN의 한계를 개선한 모델로 default로 설정된 파라미터가 많아 수정이 필요한 부분은 많지 않았다. Epoch수와 Training Image, learning rate 등을 수정하며 분석한 결과, epoch은 200회 결과와 200회 이상의 결과의 차이가 존재하지 않아 비교적 선명한 200회를 최종 파라미터로 설정하였다.

CycleGAN	
Parameter	**기준**
Training image	772장 (낮) 290 / (밤) 482
Image size	640*360
Learning rate	2e-4(G),2e-4(D)
Batch size	1
Beta	0.5(1), 0.999(2)
Epoch	20

TodayGAN	
Parameter	**기준**
Training image	1452장 (낮) 774 / (밤) 519
Image size	1318*741
Learning rate	2e-4
Batch size	1
Beta	0.5
Epoch	200

4.2 CycleGAN의 학습 안정성을 위한 구조 수정

Hyper parameter 조정 후에도 발생한 문제들 (ex. mode collapse, Generator의 loss가 지나치게 큰 현상)을 해결하기 위해 GAN학습에 대한 논문과 사례들을 조사한 후 적용하였다.

가. Generator의 학습횟수 조정

GAN을 학습시키면 Generator와 Discriminator의 loss가 번갈아 가며 감소하며 수렴하는 모습을 보여야 한다. 그러나 실제 학습을 시켰을 때 Generator의 loss가 Discriminator의 loss에 비해 크며(Generator는 3 이상, Discriminator는 1 이하) 잘 감소하지 않는 경향을 보여 GAN을 학습시킨 사례들을 조사해 보았다. 그러던 중 AnoGAN을 사용한 연구인턴4기의 프로젝트 보고서에서 Generator의 loss가 Discriminator보다 큰 문제를 해결하기 위해 Discriminator를 한번 학습시킬 때 Generator를 4번 학습시키는 방법으로 해결했다는 내용을 발견하였고 이를 적용하여 Discriminator와의 균형을 확보하고자 하였다.

나. Learning rate decay 적용

Learning rate decay는 인공신경망을 학습시킬 때 최적해를 찾기 위해 learning rate를 일정하게 유지시키는 것이 아니라 학습이 진행됨에 따라 감소시키는 것을 뜻한다.

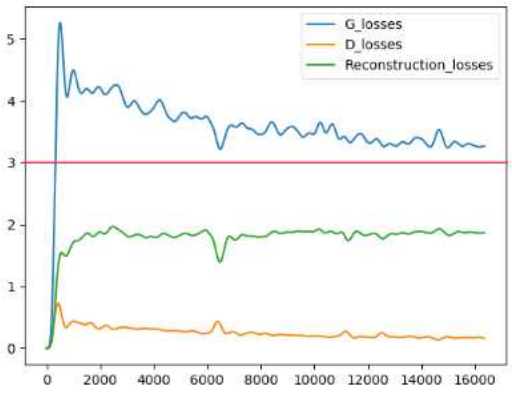
우리가 optimizer로 사용하는 Adam은 자체에 learning rate를 감소시키는 것이 적용되어있어 처음 학습시킬 때는 learning rate decay를 적용하지 않았으나, 조사결과 Adam에 추가적으로 learning rate decay를 적용한 결과 수렴이 잘 되어 학습 빠르게 된 사례를 발견하였다. 그렇기 때문에 적용 전과 후를 비교한 후 더 적합한 것을 최종 모델에 적용하기로 결정하였다.

다. One-sided label smoothing 적용

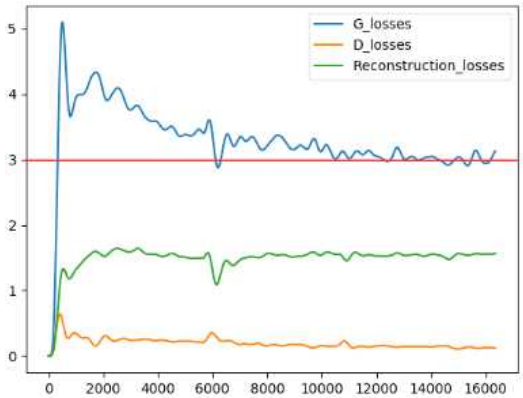
one-sided label smoothing이란 GAN을 학습시킬 때 실제 데이터에 대한 target 값을 1에서 1보다 약간 작은 값(대부분 0.9)으로 대체하여 Discriminator가 Generator의 공격을 효율적으로 대응할 수 있게해 Adversal network의 취약성을 보완하는 것이다. GAN의 개발자 Ian Goodfellow가 저술한 GAN을 학습시키는 TIP에 대한 논문(Improved Techniques for Training GANs)을 참고하였다.

Generator의 학습횟수를 증가시킨 것과 마찬가지로 Discriminator의 loss에 비해 Generator의 loss가 잘 감소하지 않는 문제를 해결하기 위해 적용하였으며 epoch=10에서 시험적으로 적용한 결과 Generator의 loss가 감소한 것을 확인하여 최종모델에 적용하기로 결정하였다.

1.0 적용



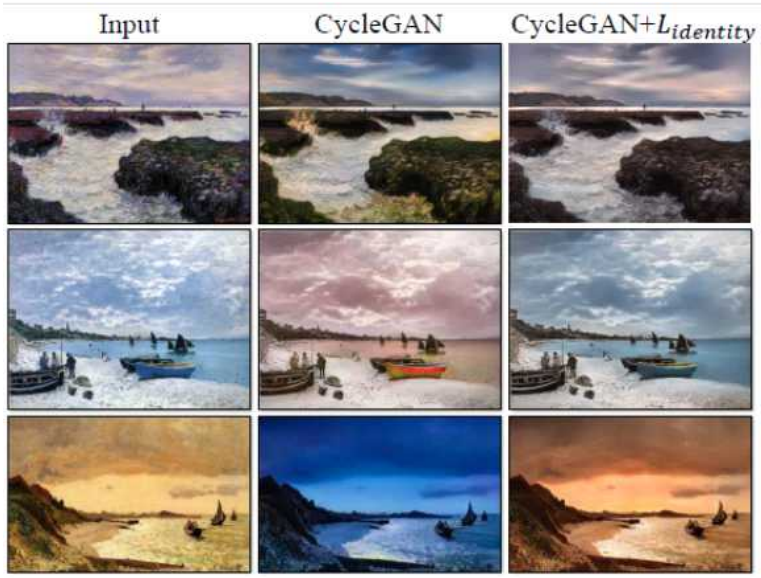
0.9 적용



라. Identity loss 추가

Identity loss란 이미지를 변환할 input의 형태와 색깔을 더 고려하도록 하기 위해 X를 Y로 변환할 때의 loss에 Y가 input으로 들어온 경우에도 Y로 변환되도록 하는 loss 추가하는 것이다.

$$\mathcal{L}_{\text{identity}}(G,F) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(y)-y\|_1] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(x)-x\|_1].$$



이미지 변환을 한 후에 원래의 이미지에 없던 물체를 생성해 내거나 색이 다르게 변환되는 경우가 생기는 것을 막기 위해 적용하였으며 시범적으로 epoch=10에서 실험한 결과 적용 후에 전보다 원본의 형태와 색을 더 잘 구현하는 것을 확인하여 최종 모델에 적용하기로 결정하였다.

마. multiscale discriminator 미적용

적용 시 Discriminator의 information을 더 반영하기 위해 Generator에 encoding/ decoding하는 과정을 추가한다. 그렇지만 적용한 두 케이스 모두 사진 퀄리티가 좋지 않고 전보다 Generator와 Discriminator의 loss도 증가했으므로 사용하지 않기로 결정하였다.

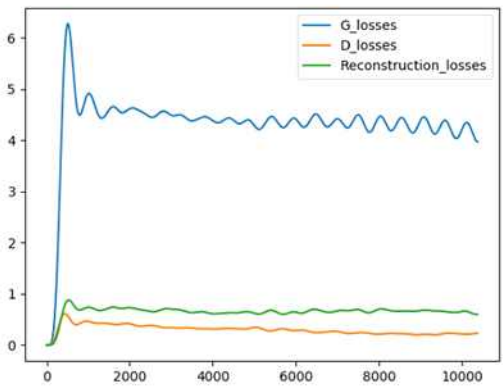
4.3 최종 결과

Epoch 100회를 기준으로 각 모델별로 학습한 결과 CycleGAN에 비해 TodayGAN의 학습결과가 훨씬 성공적인 것을 확인 할 수 있다. TodayGAN의 경우 CycleGAN 모델을 기반으로 한계점을 개선한 최근(2019)의 GAN 모델이므로 동일한 input에서 더 좋은 결과를 얻을 수 있는 것으로 보인다. 각 모델별 Loss 그래프에서도 TodayGAN의 Generator Loss가 0.4 정도에 수렴되는 결과를 확인할 수 있다.

원본 CycleGAN

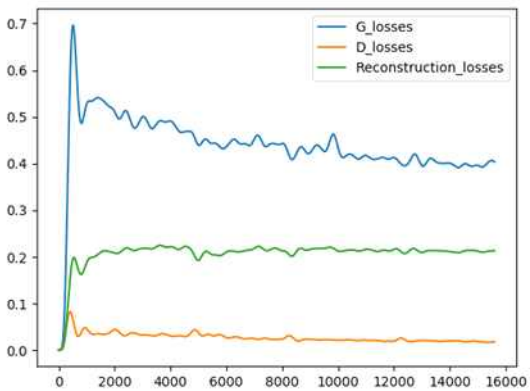
TodayGAN

CycleGAN



Generator Loss 4
Discriminator Loss 0.5
Reconstruction Loss 1

TodayGAN



Generator Loss 0.4
Discriminator Loss 0.2
Reconstruction Loss 0.05

5. 드론

5.1 드론 선정사유

드론의 종류에는 코딩이 가능한 드론과 코딩이 불가능한 드론이 존재한다. 前기수 자료조사를 참고하여 코딩이 가능한 Parrot 社의 드론을 선정하였다.

Parrot 社의 제품에는 실내 드론과 실외 드론이 존재한다. 실내 드론은 GPS인식이 불가능하기 때문에 실외용 드론을 선정하였다. 실외드론에는 Bebop1, Bebop2, Anafi가 존재한다. 가장 최신 기종인 Anafi를 선정하고자 했으나 SDK가 존재하지 않아 code support를 받을 수 있는 Bebop2를 선정하였다.

5.2 드론 환경설정

Github : https://github.com/amymcgovern/pyparrot		
SDK	!pip install pyparrot (latest version 1.5.21 releases 10/16/2019)	
		from pyparrot.Bebop import Bebop
		from pyparrot.DroneVisionGUI import DroneVisionGUI

Image	!pip install opencv-python==4.1.2.30
	Import cv2

참고링크 : http://ubuntuhandbook.org/index.php/2018/05/install-vlc-3-0-2-ubuntu-16-04-ppa/	
Video Viewer	(vlc version 3.0.7) sudo add-apt-repository ppa:jonathonf/ffmpeg-3 sudo add-apt-repository ppa:jonathonf/vlc-3 sudo apt-get update sudo apt-get install vlc

Python	Version 3.6

tensorflow	!pip install tensorflow==1.15.0 !pip install tensorflow-gpu==1.15.0

5.3 드론 장거리 무선통신 환경 구축

가. 준비물

- Parrot Bebop2
- Hwawei E3372 4G USB modem
- USIM
- OTG cable
- WIFI 공유기

Figure 1. 준비물

나. 환경 구축 방법

1) 화웨이 모뎀 활성화

- 한국어로 된 동글 셋팅 페이지는 **아래링크**를 참고한다.
(http://blog.naver.com/PostView.nhn?blogId=artist_sh&logNo=221345326171&parentCategoryNo=&categoryNo=53&viewDate=&isShowPopularPosts=true&from=search)
- 화웨이 모뎀에 USIM을 장착한다.
- PC(워크스테이션)에 동글USB(=화웨이 모뎀)를 꽂는다.
- https://github.com/uavpal/beboptwo4g/wiki/Installation에 접속 후 software 부분을 읽는다.
- http://192.168.0.1/html/home.html 를 들어간다.
(이 페이지에 접속이 불가능하면 수동으로 드라이버 설치 필요)

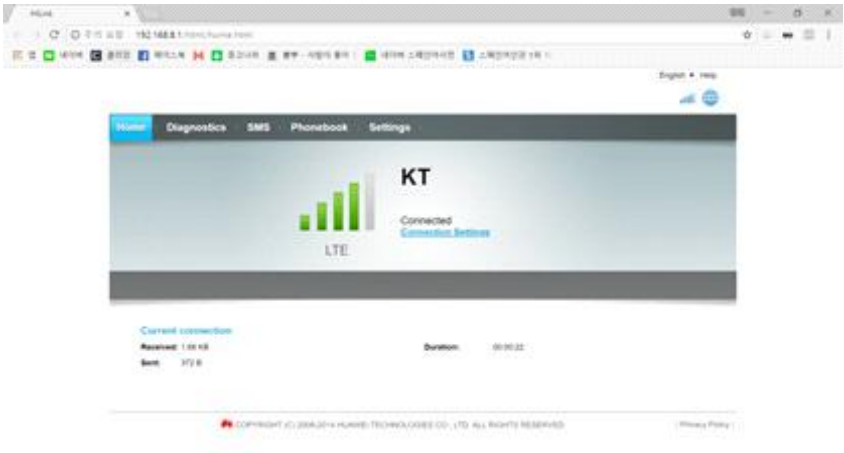


Figure 2. 정상적으로 동작하는 페이지

- 정상동작 페이지에서 setting - dial-up - profile manage 들어간다.
- profile manage에서 APN설정. (본 기수는 u+ 통신사 사용)
  (인터넷에 '통신사 APN' 치면 나온다. SKT, KT, U+ APN이 각각 있다. 이 키워드로 검색한다)
- 192.168.0.1에서 인터넷하고 연결되었는지 확인하고 넘어간다.
- 연결되면 유선으로 연결됐다고 든다. 꼭 유선이라는 것이 중요하다.

Figure 3. 유선 연결된 동글

2) Zerotire(VPN) 설정

- https://github.com/uavpal/beboptwo4g/wiki/Installation에 접속 후 ZeroTier부분을 읽는다.
- https://my.zerotier.com 방문 후,
  1. 계정 생성, 2. 네트워크 페이지 클릭, 3. 네트워크 생성을 누른다.

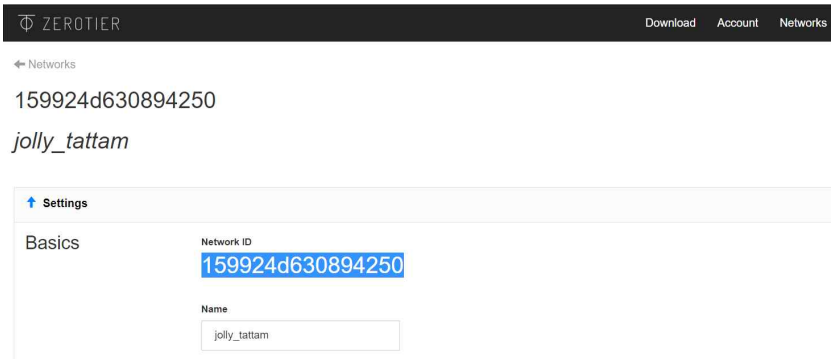


Figure 4. 정상적으로 등록된 ZeroTier 네트워크

- Network ID를 기억해 둔다. 드론에 동글 key 값을 넣을 때 사용된다.

3) 드론에 동글 Key 값 설정

- https://github.com/uavpal/beboptwo4g/wiki/Installation에서 software단락의 Download.zip을 받는다. (beboptwo4g-1.3 이라고 다운 받아짐)
- 압축을 풀고 bebop2/uavpal/conf/apn에 화웨이 모뎀 활성화에 썼던 APN을 써준다.
- bebop2/uavpal/conf/zt_networkid 에 zerotier 네트워크 ID를 입력해준다

Figure 5. APN 파일 수정

Figure 6. zt_networkid 파일 수정

- 수정된 파일을 드론에 전송하기 위해 파일질라를 설치해준다.
- Sudo apt-get install filezilla 로 파일질라를 설치한다.
- vsftpd.conf 파일설정을 바꿔준다.(https://nov19.tistory.com/64 참고)
- vsftpd.conf 파일에서 standalone = YES, ipv6 = NO 로 설정하고 나머지는 블로그에 나와 있는 대로 한다.
  (*주의 : #으로 앞에 막혀 있는거 풀어줍시다. 그리고 allow_chroot를 추가해준다)
  (**이유 : .conf 파일에서 #은 주석을 의미한다. #을 제거하면 주석이 코드로 바뀐다)
- filezilla app 의 Host에 192.168.42.1 만 찍고 quickconnect 클릭한다.

그럼 로그 창에 logged in 하고 '/' successful 뜨면 접속 성공.

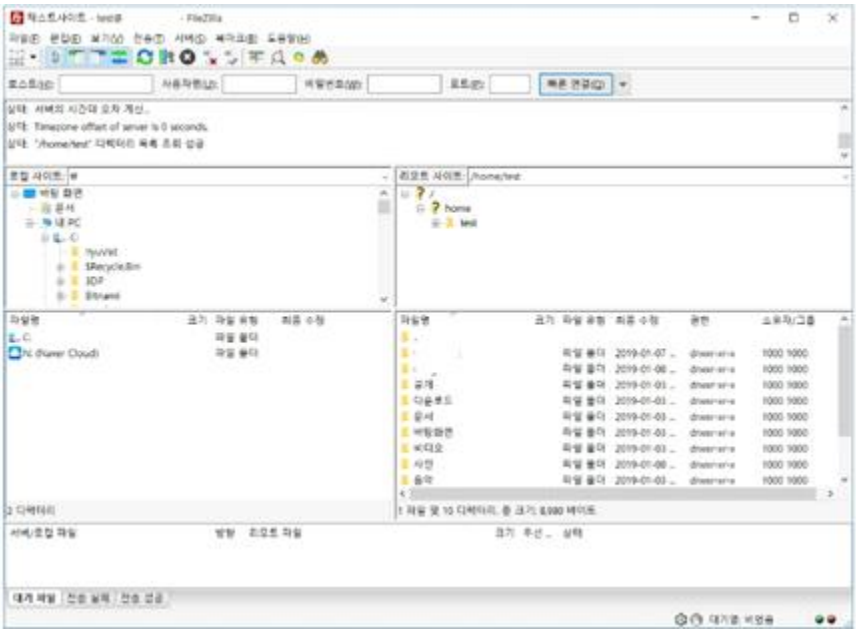


Figure 7. 파일질라 창 띄웠을 때

- 파일질라 전체창으로 하고 Transfers → FTP: File Types → Default transfer type → change from Auto to Binary. 설정한다(확인하고 넘어갈 것)
- 파일질라 왼쪽 하단 창에서 보낼 파일(폴더) 선택하고 우 클릭 후 upload 누르면 전송된다. 전부 다 전송됐는지 최 하단 창(전송성공)으로 확인한다.
- 드론 전원버튼을 빠르게 4번 눌러서 텔넷을 활성화 시킨다.
- 우분투 워크스테이션에서 cmd 창에 telnet 192.168.42.1을 명령한다./ #표시(shell)가 뜨면 접속된 것이다.
- telnet에서 아래의 명령어를 사용한다.

  mv /data/ftp/internal_000/beboptwo4g-* /tmp/beboptwo4g
  chmod+x/tmp/beboptwo4g/*/*_install.sh
  /tmp/beboptwo4g/bebop2/bebop2_install.sh
  reboot
- 정상적으로 되었다면, bebop이 켜지면서 동글에 파란불빛/초록불빛이 들어온다. 그리고 제로티어에 자동으로 등록이 된다. 확인하도록 한다.

본 기수의 경우 OTG 케이블 불량으로 자동 등록이 안 된 문제가 있었는데, 케이블을 교체하여 성공하였다.

Figure 8. 정상적으로 등록된 드론

- 정상 등록 후 제로티어에서 Allow 이더넷 브리지를 클릭해준다.

Figure 9. Allow 이더넷 브리지

4) 우분투 워크스테이션을 VPN에 등록

- 우분투(PC,워크스테이션)에서 제로티어를 설치한다.
- 우분투에 제로티어 설치 명령어 : curl -s https://install.zerotier.com | sudo bash
- 우분투에서 연결법: sudo zerotier-cli join 〈network_id〉
- 정상적으로 연결된 경우 200 join ok 메시지가 출력된다.

※ 참고한 페이지 ※

https://www.uas4g.com/downloads/uas4g-user-guide.pdf

http://blog.naver.com/PostView.nhn?blogId=hanmo&logNo=221351678168

5.4 프로젝트 구조도

드론 main코드와, GAN 모델 코드, 낮 이미지 출력, 3개의 process를 사용한다. 3개 프로세스에 대한 코드 설명을 첨부한다.

가. 드론 Main.py

User defined function name	역할	SDK module 內
userVision.savePic()	영상촬영	DroneVisionGUI._buffer_vision()
SDK 사용	드론비행	Bebop.Fly_direct()

1) 영상촬영

userVision.savePic() 함수는 DroneVisionGUI 모듈의 _buffer_vision()를 직접 수정하여 사용하였다. 수정한 부분을 아래에 첨부한다.

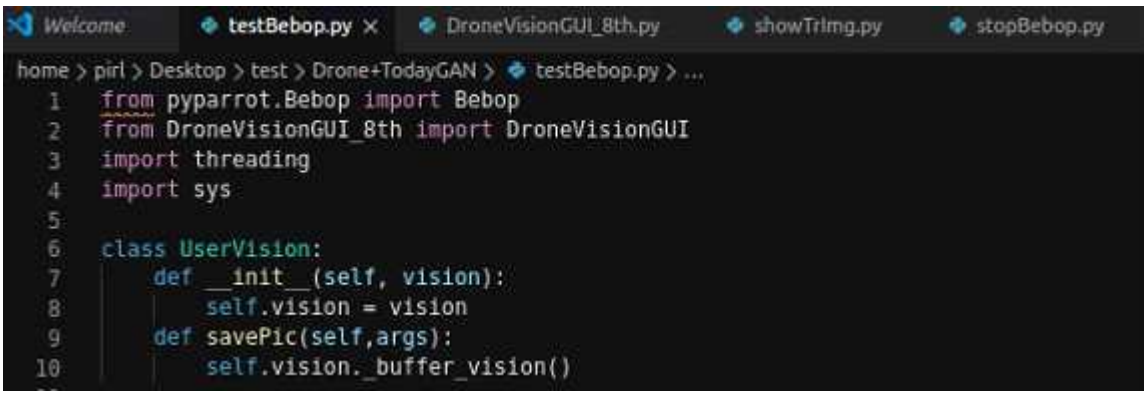


Figure 1. Main.py 內 savePic 함수 (Main.py == testBebop.py)

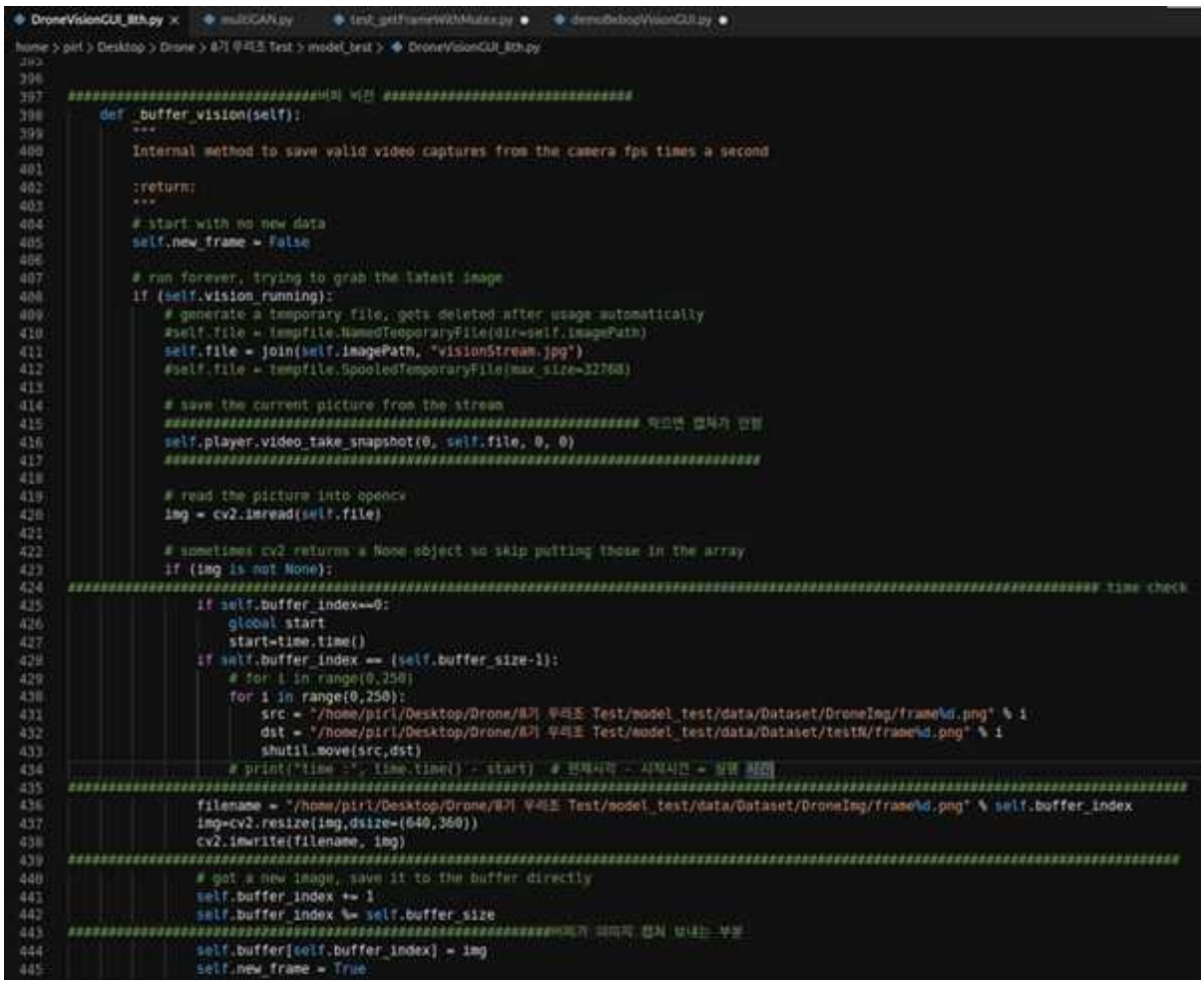


Figure 2. DronVisionGUI Module 內 _buffer_vision 함수

2) 드론비행

```
if (success):
    bebop.safe_takeoff(5)

    bebop.fly_direct(roll=0, pitch=0, yaw=0, vertical_movement=35, duration=7.5)

    bebop.fly_direct(roll=0, pitch=30, yaw=0, vertical_movement=0, duration=10)

    bebop.safe_land(10)

    bebopVision.close_exit()
    sys.exit()

else:
    print("Error connecting to bebop.  Retry")
```

Figure 3. Main.py 內 드론비행 코드

드론 비행 코드는 Pyparrot SDK github의 Example code를 참조하였다.
(github : https://github.com/amymcgovern/pyparrot/blob/master/examples/demoBebopDirectFlight.py)
사용된 SDK 함수에 대한 설명은 다음과 같으며 공식 문서 링크를 아래에 첨부한다.
(공식문서 : https://pyparrot.readthedocs.io/en/latest/bebopcommands.html#takeoff-and-landing)

2-1. safe_takeoff(timeout)
이 함수는 timeout 시간 내에 이륙(takeoff)하도록 한다.

2-2. fly_direct(roll, pitch, yaw, vertical_movement, duration)
이 함수는 드론의 진행 방향을 결정하며 roll, pitch, yaw는 항공분야의 개념이며, vertical_movement는 수직방향으로 얼만 큼 이동할지 결정하고, duration은 몇 초 동안 움직일지 결정하는 parameter이다. 이와 관련한 시각자료와 그 링크를 첨부한다.

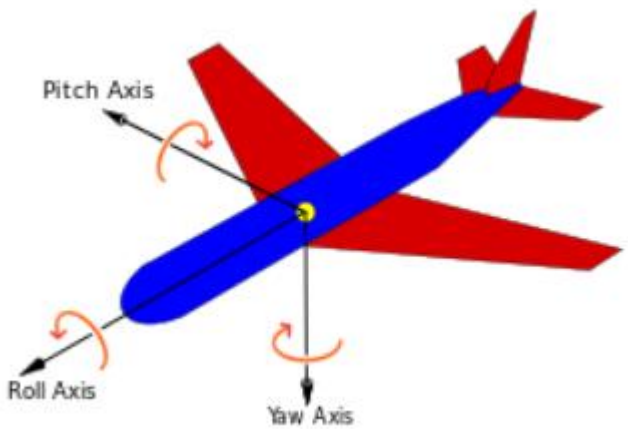


Figure 4. roll, pitch, yaw
(※참고 : http://imsjhong.blogspot.com/2017/07/yaw-pitch-roll.html)

2-3. safe_land(timeout)

이 함수는 착륙(land) 하는 함수이며 timeout 시간 내에 동작하도록 구성돼 있다.

나. GAN 모델 코드

본 코드는 github.com/AAnoosheh/ToDayGAN 에 기재된 내용을 사용하였으며 드론에 적용 시에는 test.py를 사용하였다.

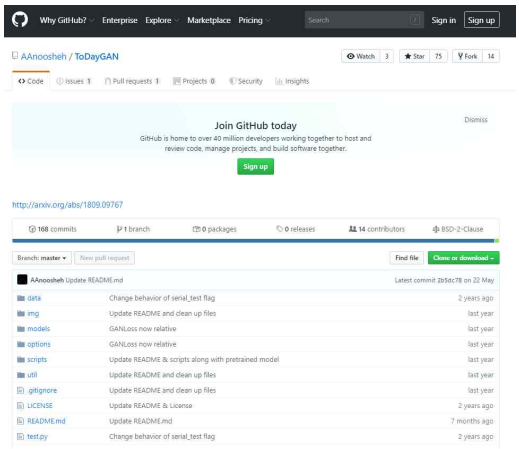


Figure 5. ToDayGAN Github Page

test 모델을 사용하기 위해선 '학습된 모델의 parameter가 기록된 파일'이 필요하다. 또한 해당 파일을 저장하기 위한 '폴더를 생성' 해야 하며 이에 대한 설명은 다음과 같다.

1) 그림과 같은 경로에 checkpoints-night2day 폴더를 생성한다.

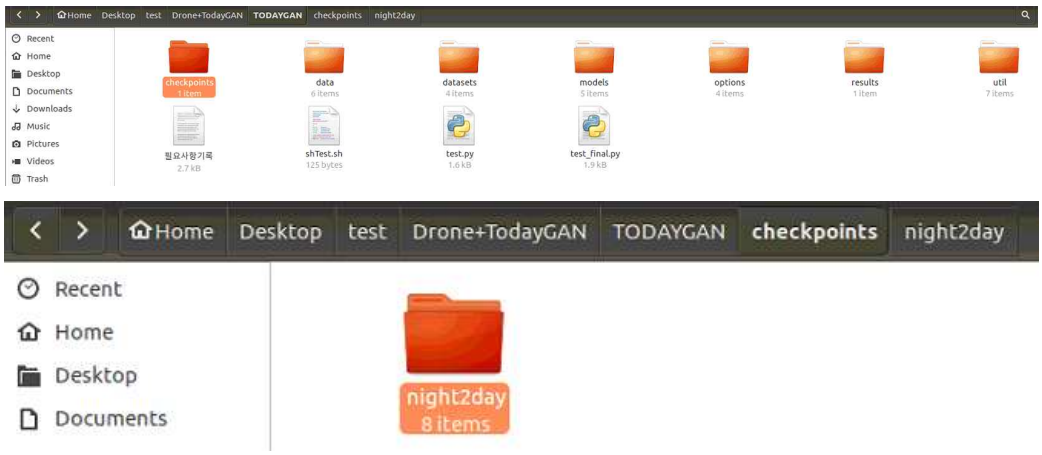


Figure 6. 폴더생성 위치와 경로

2) 해당 위치에 200_net_D0.pth 와 같은 파일을 배치한다.

200_net_D0.pth 파일은 학습(train)시에 자동으로 생성되며, pth확장자가 parameter가 기록된 파일이다. 200은 epoch 수를 의미하고 D와 G는 각각 GAN의 기본 구조인 Discriminator와 Generator를 의미한다. D0와 D1에서 0과1은 Discriminator의 숫자를 의미한다. 0과 1밖에 없는 본 경우는 2개의 Discriminator가 존재한다는 의미이다.

Figure 7. pth 파일

다. 낮 이미지 출력

낮 이미지 출력은 showTrimg.py 라는 파일에 소스코드가 저장돼 있다. opencv를 사용하여 image file을 읽고, for 문을 사용하여 변환된 50장의 이미지(=frame)만을 cv2.imshow로 출력하였다.

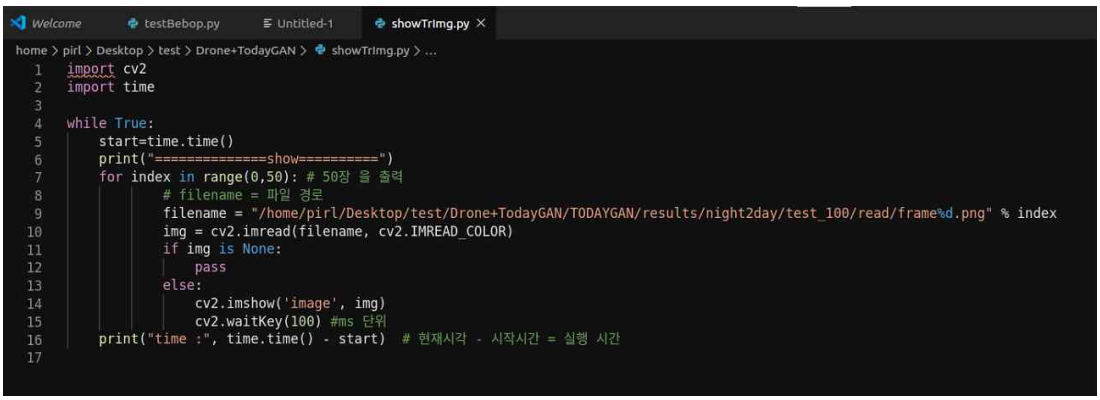


Figure 8. showTrimg.py 파일

5.5 프로그래밍 이슈

가. Multi-thread Issue

VLC를 사용하는 데모코드 dls demoBebopVisinGUI.py 에서
(데모코드 링크 :https://github.com/amymcgovern/pyparrot/tree/master/examples)
set_user_callback_function을 제외한 Thread는 동작하지 않는 문제 발생했다.
먼저, set_user_callback_function 은 Thread를 생성하는 함수와 동일하다.

bebopVision.set_user_callback_function(userVision.usrfunction, user_callback_args=None)
threading.Thread(target=usrfunction)

〈SDK 함수와 동일한 Python 내장 함수〉

구조를 파악한 결과,

```
# from pyparrot.DroneVisionGUI import DroneVisionGUI
from DroneVisionGUI_8th import DroneVisionGUI
```

DroneVisionGUI라는 Module에서

```
def _start_video_buffering(self):
    """
    If the video capture was successfully opened, then start the thread to buffer the stream

    :return: if using libvlc this will return whether or not the player started
    """
```

Figure 1. _start_video_buffering 함수

_start_Video_buffering이라는 function의 app.exec()를 주석으로 처리하면 멀티쓰레드가 가능하게 해결된다. 그 결과 아래에 보이는 Main 함수에서 MultiThread를 사용할 수 있다.
(trImgshow= usr defined function)

```
if (success):
    bebopVision = DroneVisionGUI(bebop, is_bebop=True, user_code_to_run=demo_user_code_after_vision_opened,
                                 user_args=(bebop, ))#, user_draw_window_fn=draw_current_photo)


    userVision = UserVision(bebopVision)
    bebopVision.set_user_callback_function(userVision.savePic_GAN_imShow, user_callback_args=None)
    # set user call back function 은 open_video 활성화 되었을 때 적용된다.
    bebopVision.open_video()

    #되는거
    IMGShow_Thread=threading.Thread(target=trImgShow_tread).start()
```

Figure 2. Main 함수에 thread 적용 가능 상태

나. Transaction Conflict Issue (Read/Write)

동일한 파일에 읽기와 쓰기를 동시에 접근하면 error가 발생하는 문제가 있었다. 이를 해결하고자 read할 파일이 있는 폴더와, write하는 파일의 폴더를 구분하여 사용하였다. 경로를 설정하고 수정하기 위해, 드론 카메라가 촬영하는 Capture Frame이 Code 내에서 저장되는 부분을 찾았다.

```
DroneVisionGUI_8th.py ×  multiGAN.py  test_getFrameWithMutex.py ●  demoBebopVisionGUI.py ●
home > pirl > Desktop > Drone > 8기 우리조 Test > model_test > DroneVisionGUI_8th.py
398     def _buffer_vision(self):
399         """
400         Internal method to save valid video captures from the camera fps times a second
401
402         :return:
403         """
```

Figure 1. DroneVisionGUI 파일 內 _buffer_vision 함수

DroneVisionGUImodule내의 _buffer_vision 함수에서 self.buffer[] 배열에 캡쳐된 이미지가 저장된다.

Figure 2. _buffer_vision 함수 수정 내역

(※ 주의 : shutil.move 를 사용하기 위해서는 import shutil 이 필요하다. Import shutil을 위해서는 pip install shutil을 하면 된다.)

6. 기대효과 및 보완사항

1. 기대효과

1) 사각지대 및 범죄 취약지역의 야간순찰 기능

기존의 야간순찰은 고정된 CCTV를 이용하거나 사람이 직접 순찰하는 방법으로 순찰이 이루어졌다. 하지만 고정형 CCTV는 촬영 범위에 한계가 존재하며, CCTV의 위치나 각도에 따라 사각지대가 발생한다는 문제점이 존재한다. 또한 CCTV는 야간에 사람 및 사물에 대한 식별 능력이 떨어지는데, 대부분의 범죄가 야간시간대에 발생하는 점을 고려하면 야간 감시기능이 미흡한 CCTV는 반드시 개선이 필요하다고 할 수 있다.

이러한 문제점을 해결하고자 동 프로젝트에서는 드론을 이용하여 야간영상을 주간영상으로 변환이 가능한 이동형 CCTV를 구현하였다. 동 프로젝트에서 구현한 이동형 CCTV를 활용할 경우, 사각지대 및 범죄 취약지역의 경우 실시간 순찰이 가능하며, 야간영상이 주간영상으로 변환되므로 사람 및 사물에 대해 식별이 용이할 수 있다. 이를 통해 사전에 발생할 수 있는 범죄를 예방하고, 고정형 CCTV의 한계를 극복할 수 있다.

2) 재난재해 상황 시 야간 수색 가능

최근 들어 대규모 산불 등 예기치 못한 재난재해가 발생하는 상황이 발생하고 있다. 재난재해가 야간에 발생 할 경우 인명피해를 최소화하는 것이 무엇보다 중요하다. 지금까지는 재난재해 상황 시 수색이 어려운 경우에는 다음 날 동이 틀 때까지 기다리는 것이 유일한 방법이었다. 동 프로젝트에서 구현한 모델을 바로 적용하는 것은 무리가 있으나, 산, 바다 등 다양한 야간상황에 대한 학습이 사전에 이루어 질 경우 재난재해 상황에서도 야간 수색이 가능하다.

2. 보완사항

본 프로젝트의 드론을 통해 야간순찰을 진행하며 밤 영상을 낮 영상으로 실시간 변환하는 모델을 구현하고자 하였으며, 초기 설정한 프로젝트의 목적을 충분히 달성하였다.

밤 영상을 낮 영상으로 변환하는 학습모델 구현, 드론을 통해 실시간으로 밤 영상이 낮 영상으로 변환되는 목표의 구현은 성공하였으나, 성능 부분의 개선이 필요한 점이 필요한 것이 한계로 남는다.

그 외 다음과 같은 개선점들이 존재한다. 첫째로 밤 영상을 낮 영상으로 변환 시 object detection을 통해 사람과 차량을 구분하는 기술에 대한 구현이 필요하다. 교수님의 조언에 따라 해당 기술까지 구현해보려고 시도하였으나, 시간적인 한계로 시도에 그쳤다. 두 번째로 드론의 완전한 자율주행 기술이 필요하다. 코드를 통해 드론의 자율주행이 일부 가능하지만, 원하는 구간을 모두 탐색하거나 순찰하는 부분은 일부 개선이 필요하다. 세 번째로 드론의 장애물 회피 기능 구현이 필요하다. 자율주행을 가능하게 하기 위해서는 드론 스스로 장애물을 인지하고 경로를 찾아가는 기능이 구현되어야 하는데, 다음 기수에서는 이 부분에 대한 해결이 이루어지면 드론으로 더 많은 기능을 구현할 수 있을 것 같다.

7. 팀원소개와 상호평가

1. 팀원 상호평가

공해인

[본인소감] 먼저 3개월 간 함께 수고해준 팀원들 모두 고맙습니다. 교육 기간 동안 힘들고 버거웠던 순간도 있었지만 옆에서 도와준 팀원들 덕분에 지금까지 올 수 있었던 것 같아요. 끝이 올까 싶었는데 정말 오긴 오네요. 이제 각자의 삶으로 돌아간다고 생각하니 시원섭섭합니다. 앞으로도 우리 팀원 모두 진심으로 응원할게요 :)

- 항상 책임감을 가지고 팀을 이끌며 자기 맡은 일뿐만 아니라 모델학습팀에도 큰 도움을 준 만능조장이었습니다.
- 조장으로서 팀을 잘 이끌어 주었습니다.
- 조장으로서 조원들을 살필 뿐 아니라, 도움이 필요한 부분에 어디든 도움을 주는 만능테이너 해인:)
- 어떻게 팀에 기여할지 끊임없이 고민하는 팀원이었습니다. 드론과 GAN모델 , 전처리, 영상편집, PPT제작까지 프로젝트 전반에 걸쳐 모델팀과 엔진팀을 넘나들며 모든일에 기여를 했습니다. 특히나 모델을 선정하는 과정에서 선행논문을 분석하고 파악하였으며, 자신이 가장 잘 할 수 있는 일에서 뛰어난 역량을 기반으로 높은 성과를 보여주었습니다. 더불어 파이썬을 처음 배웠음에도 불구하고 전처리에 필요한 코드를 작성하는 능력까지 보여주었습니다. 또한 영상편집 역량까지 겸비하여 적재적소에 필요한 포지션에서 자신의 할 일을 찾아나서는 뛰어난 팀원이었습니다.

김소희

[본인소감] 드론을 다뤄본 경험도 모델학습도 처음이라 많은 어려움이 있었지만, 기대했던 목표를 충분히 달성하여 만족스러운 경험이었습니다.

- 최종 모델 완성하기 위해 수없이 학습시킬 정도로 노력하고 보고서도 완벽하게 만드는 능력자였습니다. 옆자리에서 빅데이터프로젝트 때부터 많은 도움받았는데 AI프로젝트를 하면서도 혼란스러워하는 저를 이끌어주어서 고마웠어요.
- PPT 및 보고서 여신. 프로젝트 동안에 ppt와 보고서 중 이분의 손을 안거쳐 간 것들이 없다.
- 소희누나는 항상 양보를 많이하는 팀원이었습니다. 자기 주장 보다는 항상 팀을 위해 희생을 많이했고 이런 저런 여러 일을 맡을 때도 단 한번도 불평없이 모두 감수했습니다. 또한 항상 팀원들을 많이 챙겨주었으며 ppt , 보고서 등 주 업무 외에도 필요하지만 메인이 아닌 일들도 많이 해줘서 정말 고마웠습니다. 그리고 누나랑 의견이 달라서 부딪힐때도 짜증 한번 안내줘서 정말 고맙고 미안했습니다. 그 외에도 누나가 일하다 오셔서 업무에 대한 노하우로 노련하고 숙련된 솜씨로 맡은 일을 처리해주셨고 논문을 잘 보셔서 이슈가 발생했을 때 관련된 논문을 검색하고 해결책을 찾는 등 같은 조여서 정말 든든했습니다.
- 빅데이터 프로젝트에서 보고서 제작을, AI 프로젝트에서는 모델 학습을 담당한 팀원입니다. 매 프로젝트마다 열심히 맡은 일을 잘 마무리해줘 든든하고 고마웠습니다. 특히 영혼을 담아 만든 보고서는 정말 감동이었습니다. 수업 태도도 아주 좋았는데, 어렵고 힘든 수업도 항상 집중하며 듣는 모습이 기억에 남습

니다.

박다연

[본인소감] 파이썬도 어렵고 드론도 너무 어려운 주제같았는데 능력있는 팀원들 덕분에 잘 마무리할 수 있었습니다. 다들 이 이후에도 원하는 결과 얻었으면 좋겠고 항상 고마움을 잊지않을게요:)

- 코딩천재 더이상의 말은 필요 없다.
- 항상 묵묵하게 주어진 일을 해내고, 어려운 부분이 생겨도 끈질기게 해내는 코딩천재 다연:)
- 묵묵하게 맡은 일을 성실히 수행하고 항상 아웃풋을 가져오는 팀원이었습니다. 모델팀에서 모델에 관한 다양한 논문을 찾아보는 사전 조사력을 보여주었으며, 파이썬을 처음 배웠음에도 불구하고 모델을 사용하기 위한 코드를 분석하고 수정하는 역량까지 보여주었습니다. 모델의 성능이 좋지 않은 이슈가 발생했을 때 더 나은 모델을 찾기위해 서칭을 하고 모델간 특징 및 차이점을 잘 발굴해 내었으며, 성능향상을 위해 어떤 파라미터를 조정해야 하는지 끊임없이 고민을 하였습니다. 뿐만 아니라 조사한 내용을 보고서로 깔끔하게 정리하는 능력을 보여주었으며, 자신이 시도한 내용과 결과를 글로 적어 공유하는 전달력까지 겸비하였습니다.
- 옆자리에서 제 질문을 많이 받아준 고마운 팀원입니다. 질문을 하면 친절하게 알려주고, 문제가 생기면 해결해주려고해서 고마웠습니다. AI 프로젝트에서는 모델 학습을 맡아서 모델 공부도 열심히 하고, 코드 분석도 성실히 수행하였습니다. 조용하지만 성실하게 맡은 일을 수행하는 팀원입니다. 개인적으로는 같이 산책하며 이런저런 얘기 나눠줘서 고마웠습니다.

성준혁

[본인소감]

- AI 프로젝트장으로서 남다른 애정을 가지고 드론과 프로젝트에 올인. 가장 고생이 많고 AI에 열정이 넘치고 팀원들을 위해 노력한 드론파파 칭찬합니다.
- AI 프로젝트장으로서 드론과 모델학습을 모두 아우르고 누구보다 가장 많은 고민을 했던 듬직한 드론파파 준혁:) 주어진 모든 일을 완성도 있게 추진하기 위해 많은 시간 고민하고 수많은 실행을 통해 반복, 개선해나가는 모습이 같은 조원으로서 너무나도 든든했습니다.
- 팀프로젝트조장을 맡아 가장 힘들었을 것 같아요. 끝까지 프로젝트완성을 위해 코딩과 전반적인 부분을 신경쓰느라 고생많았습니다. AI에 대한 관심이 큰 것이 보였는데 앞으로도 AI분야에서 성공하길바래요.
- 프로젝트 매니저로서의 책임감으로 항상 프로젝트에 대해 고민이 많은 팀장이었습니다. 밥먹는 시간에도 프로젝트에 대한 고민을 놓지않을 만큼 집요함과 열정을 보여주었고 매일 늦게까지 남아 프로젝트를 하는 끈질김을 가지고 있었습니다. 또한 훌륭한 코딩실력을 갖추고 있었고 환경설정에 대한 깊은 탐구와 고민으로 팀원 전체의 환경설정까지 도맡아 진두지휘하는 역량을 보여주었습니다. 하나를 잡으면 끝까지 깊이 파고드는 성향과 학문에 대한 많은 호기심을 가진 성준혁 팀장은 연구분야에 높은 적합성을 보이며 향후 대학원진학을 해서도 충분히 잘할 것이라 믿어 의심치 않습니다.
- AI 프로젝트장이자 드론파파를 맡아 프로젝트를 리드한 팀원입니다. 적극적으로 프로젝트 리드를 해주었고 최종 발표도 맡아서 본인 역량의 120%를 수행했다고 생각합니다. AI 프로젝트에서 혼자 짐을 많이 진 것 같아서 미안하고 고마운 마음이 있습니다. 빅데이터 프로젝트에는 데이터 전처리(파생변수생성)

와 보고서, 웹사이트 제작 등의 작업을 성실하게 수행했습니다. 어떤 프로젝트에 참여해도 적극적이고 성실하게 임할 조원입니다. 마지막으로 특유의 위트와 재치로 교육 기간 동안 팀원들을 즐겁게 해주었습니다. 저도 덕분에 많이 웃었습니다. 고마워요!

이건우

[본인소감] 3개월동안 합숙생활하면서 다양한 곳에서 온 사람들과 오랫동안 부대낄 수 있는 시간이었습니다. 밤낮으로 고생한 팀원들에게 감사하며 끝까지 함께하느라 수고가 많았다고 전해주고 싶습니다. 인적성 시험, 면접 등 여러 사유로 자리를 비운 적이 많아 프로젝트에 많이 참여하고 기여하지 못해서 미안한 마음이며 , 팀프로젝트 기간에 예민하게 굴어서 팀원들의 사기에 영향을 끼친 점도 미안하게 생각합니다. 여러모로 팀원들에게 미안한 마음이 가장 크게 남으며, 그동안 데리고 같이가느라 고생많았고 고마웠습니다.

- 이분 앞에선 불가능은 없다. 뭐든 해결해내는 인재.
- 올빼미 생활을 밥먹듯이하며 새벽 늦은시간까지 드론과 학습모델 구현에 힘쓴 코딩전문가 건우:)
- 수업들으면서도 코딩도움 많이 받았는데 프로젝트하면서도 코딩실력에 감탄했습니다. 모두들에게 새로워서 어려워하는 드론파트를 맡아 프로젝트를 완성시키느라 수고많았습니다.
- 팀 프로젝트에서 없으면 안될 존재입니다. 빅데이터 프로젝트에서는 웹사이트 제작을, AI 프로젝트에서는 드론을 맡았고 성공적으로 해당 작업을 수행해주었습니다. 다른 팀원들은 하지 못하는 부분을 너무 잘 수행해주는 똑똑한 팀원입니다. 개인적으로는 파이썬 프로그래밍 과제와 취업 준비 등에서 도움을 받아 고마운 마음이 있습니다.

장하민

[본인소감] 관련전공자도 아니었고 새롭게 도전하는 분야였지만, 팀원들과 함께 끝까지 할 수 있었습니다. 파이썬, 빅데이터, 인공지능 까지 어려운 과목들이 많았지만, 목표를 가지고 끝까지 할 수 있어서 감사했습니다. 이 기회를 바탕으로 더욱더 발전하겠습니다. 특히 저희 팀원들 그리고 나머지 교육생을 고생하셨습니다.

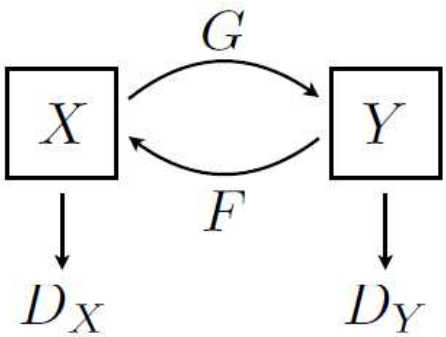
- 주어진 일은 물론 프로젝트와 관련된 모든 일에 가장 적극적으로 참여하며 리드한 열정맨 하민:)
- 맡은 일에 항상 열정적으로 임하였으며 모델학습을 위해 끊임없이 노력하는 모습을 보여줬습니다. 제가 방향을 못잡을 때 항상 실행력있게 추진하는 모델팀 팀장의 모습을 보여줘서 팀에 꼭 필요한 인재라는 생각을 하게 되었어요.
- posco 기업을 사랑하고 포항을 사랑하는 장하민 팀원은 항상 팀 분위기를 밝게 만들어 주었습니다. 유머있는 농담으로 분위기를 항상 재밌게 만들어 주었으며, 부드럽고 유연한 말솜씨로 팀 내 의견 조율과 화합을 이끌어 내었습니다. 기계공학 전문가로써 드론이 고장났을 때 직접 드론을 수리하여 드론에 새 생명을 불어넣어주었습니다. 또한 생각치도 못한 창의적이고 신선한 아이디어를 제시할 수 있는 능력까지 겸비하였으며 제시한 아이디어를 실행에 옮기는 추진력까지 보여주었습니다. 지치지 않는 열정으로 마지막까지 최선을 다하며 시연영상을 제작하는 과정에서 각본,연출,편집까지 모두 참여하는 모습을 보여주었습니다.
- 드론 수리, 영상 편집, 발표 능력 등 여러 능력을 갖춘 다재다능한 팀원입니다. 본인이 맡은 일을 성실하게 수행하는 조원이고, 특히 행동력과 실천력이 강합니다. AI 프로젝트에서 끝까지 포기하지 않고 드론을 고쳐내는 모습이 인상 깊었습니다. 또한 수업 시간마다 적극적으로 질문하는 모습이 보기 좋았습니다.

8. 선행연구

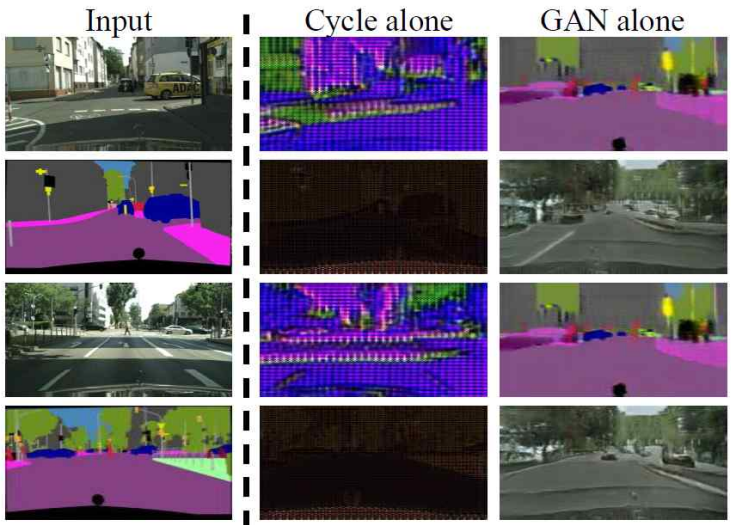
1. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (2017, Jun-Yan Zhu외)

본 논문은 Pix2Pix를 발표한 UC Berkeley의 Alexei A. Efros 교수의 연구실에서 이어서 나온 논문으로 핵심은 Unpaired Data에 있다. 이미지 변환 시 Paired Dataset으로 학습을 하면 가장 좋겠지만, 실제 세계에서는 이러한 경우는 거의 없는 경우가 대부분이다. CycleGAN에서 주로 다루고 있는 예시 중 모네의 그림을 사진으로 변환하기 위해서는 pair로 된 이미지 데이터가 없으므로(모네의 그림과 정확히 똑같은 풍경을 가진 사진을 찍는 것은 불가능함) 본 논문에서는 Cycle Consistency 로 이 문제를 해결하고자 하였다.

본 논문에서는 CycleGAN의 구조를 아래와 같이 설명하고 있다.

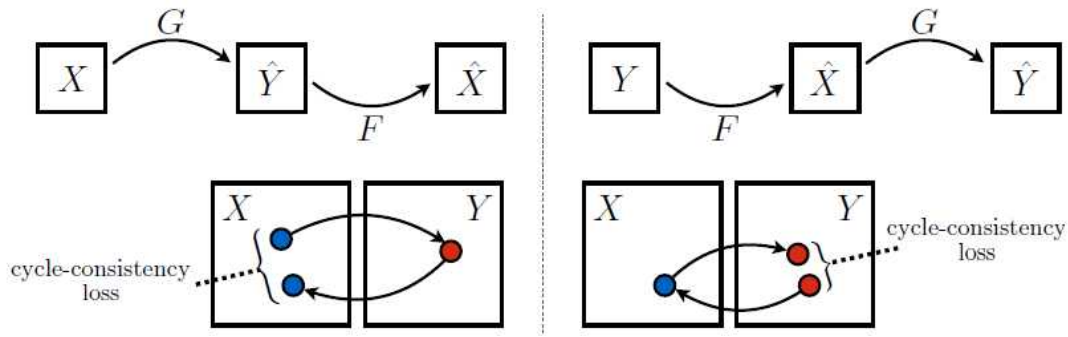


여기서 X와 Y는 우리가 연결하고자 하는 Image인데, 여기서는 다수의 Image Set라고 이해하는 것이 적합하다. X Domain에서 Y로 Mapping하는 G Function이 존재하고 이것이 진짜 Y인지 아닌지를 Y의 Discriminator를 통해서 검사를 받게 된다. 이 경우 X의 성질을 유지하면서 Y로 Style을 바꾸는 것이 아니라, Y Domain의 진짜 Image처럼 흉내내면 되기 때문에 별 의미 없는 Mapping이 되어버린다. 위 그림과 같은 구조에서는 그냥 Y Domain의 이미지처럼만 보이면 그만이기 때문에 변하기 쉬운 이미지만 만들고 끝나게 된다.



이러한 문제를 해결하기 위해 본 논문에서는 Cycle-Consistency를 적용하였다.

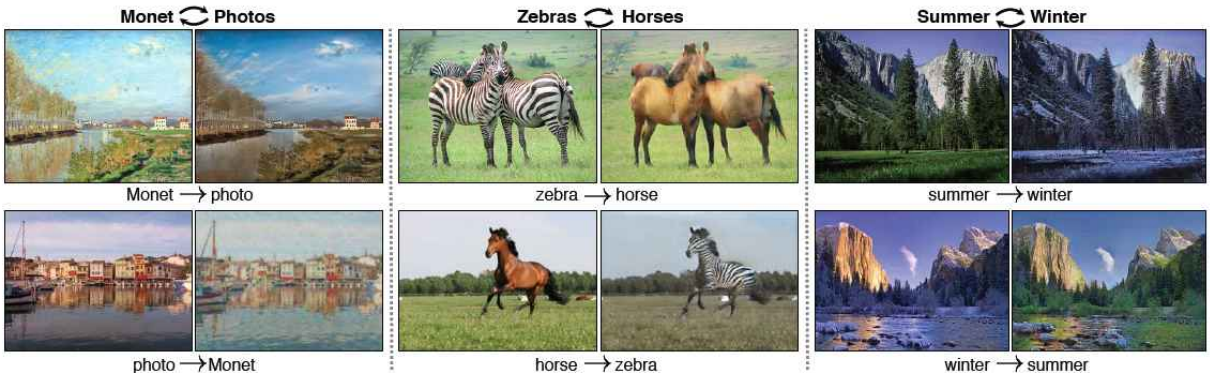
Cycle-Consistency는 단순한 Mapping이 아니라 Y로 매핑되었다가 X로 다시 돌아오는 Mapping까지 고려해서, 원래형태를 유지하며 돌아오도록 만드는 제약 조건을 걸어 주는 것이다. 이것을 반대의 Mapping에도 동일하게 적용해준다.



이렇게 하는 이유는 해당하는 Y가 우리의 Dataset에 없기 때문에(unpaired data), Y로 갈 때는 Y Domain처럼 보이는지 확인하고 실제 제약 조건은 다시 X로 돌아올 때 형태를 유지하게 만드는 것이다.

이것을 Loss Function을 통해서 보면 명확해지는데, X에서 Y로 가는 GAN Adversarial Loss에서, Cycle을 돌아서 다시 X로 돌아 오는 Loss를 추가해 주고, Y에서 X로 가는 경우에도 똑같이 적용한 다음, 이것을 더한 것을 전체 Loss로 정의해서 학습하고자 하는 것이다.

이렇게 하면 Network가 다시 돌아와야 하므로 형태를 크게 바꿀 수 없고, Adversarial Loss를 통해서 진짜 Y Domain처럼 보여야 하기때문에 Style을 Y처럼 바꾸게 되므로, 아래 그림과 같이 우리가 원하는 Style Transfer 작업을 할 수 있게 된다. 모네의 그림에서 물체들의 큰 형태들은 남긴 채 Style을 Photo Domain처럼 보이도록 만들고, 이와 동일한 원리로 말을 얼룩말로, 여름 이미지를 겨울로 바꾸는 것이 가능해진다.



2 운전자 안정성 향상을 위한 Generative Adversarial Network 기반의 야간 도로 영상 변환 시스템 (2018, 안남현 외, 한국방송·미디어공학회)

본 논문에서는 딥러닝 알고리즘인 Generative Adversarial Network(GAN)을 기반으로 야간 도로 영상을 보정하여 주간 영상으로 변환하는 알고리즘을 제안하고 있다. 영상 특성 변환 분야에서 높은 성능을 보이는 GAN알고리즘으로는 pix2pix가 대표적이나, 이는 입력영상에 대해 목적영상(저자가 부여하고자 하는 특성을 지닌 영상)이 쌍으로 존재 해야 하는 제한요소가 있으므로 이를 해결하기 위해 Cycle GAN을 적용하였다.

학습을 위해 전방 블랙박스 카메라로부터 촬영된 도로 영상을 입력받아 입력된 영상을 가로 축을 따라 세 부분으로 분할한 뒤, 일괄적으로 이미지 변환 모듈을 통해 각각 낮 영상으로 변환한다. 변환된 영상은 다시 결합된 뒤 운전자에게 제공되어 시각적 편의를 제공하는데 목적이 있다.

학습에 사용된 데이터는 총 2000장으로 야간 도로영상과 주간 도로영상 각각 약 1,000장씩을 사용하였다. 학습 횟수는 50 epoch으로 진행하였고, input resolution은 256*256으로 설정하였으며, 학습결과 야간 도로 영상이 generative network를 통해 자연스러운 주간 도로 영상으로 변환되는 것을 확인 할 수 있다.

분석결과 256 사이즈의 경우 자연스러운 변환 영상을 얻을 수 있었으나, 추후 resolution이 더 큰 영상에서도 가능한지, 좀 더 어두운 야간 영상에 대해서도 적용이 가능한지에 대한 연구의 필요성을 끝으로 제시하였다.

3. On the replication of CycleGAN (2018, Robin Elbers)

본 논문은 GAN의 개념에서 출발하여 기존의 GAN에서 발전된 pix2pix,UNIT등의 모델들에 대해 설명한 후 CycleGAN을 기반으로 성능을 향상시키기 위한 구조 수정을 제안한다.

CycleGAN은 paired 데이터가 아닌 경우에도 image-to-image translation이 가능한 인공신경망으로, reconstructed image와 원본을 비교하는 cycle-consistency loss가 있는 것이 가장 큰 특징이다. CycleGAN의 구조의 경우 Discriminator에는 pix2pix와 마찬가리조 PatchGAN을 사용하며 Generator의 경우 Resnet을 사용한다. 따라서 이 논문은 CycleGAN의 구조를 Generator를 U-Net구조로 변경하거나 adversarial loss를 Wasserstein loss로 대체하여 원본의 CycleGAN과 FCN-score로 성능을 비교하였다.

	Per-pixel accuracy	Per-class accuracy	Class IOU
Zhu et al.	0.559	0.187	0.132
Baseline (ours)	0.704	0.234	0.171
U-Net generator	0.688	0.222	0.169
No image buffer	0.720	0.237	0.179
Wasserstein loss	0.492	0.112	0.070

9. Reference

1. Night-to-Day Image Translation for Retrieval-based Localization,Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, Luc Van Gool, last revised 4 Mar 2019
   https://arxiv.org/abs/1809.09767

2. Unsupervised Image-to-Image Translation Networks,Ming-Yu Liu, Thomas Breuel, Jan Kautz,last revised 23 Jul 2018
   https://arxiv.org/abs/1703.00848

3. PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup,Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein,June 2018
   https://adoberesearch.ctlprojects.com/wp-content/uploads/2018/04/CVPR2018_Paper3623_Chang.pdf

4. Image-to-Image Translation with Conditional Adversarial Networks,Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros,last revised 26 Nov 2018
   https://arxiv.org/abs/1611.07004

5. Keras implementation of CycleGAN,simontomaskarlsson
   https://github.com/simontomaskarlsson/CycleGAN-Keras

6. Improved Techniques for Training GANs, Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, June 2016
   https://arxiv.org/abs/1606.03498

7. Mode Regularized Generative Adversarial Networks, Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, Wenjie Li, Mar 2017
   https://arxiv.org/abs/1612.02136

8. Generative adversarial nets. In Advances in Neural Infor-mation Processing Systems, Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, 2014
   https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

9. Are GANs Created Equal? A Large-Scale Study, Mario Lunic, Karol Kurach, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, 2018
   https://arxiv.org/abs/1711.10337

10. How to Train a GAN? Tips and tricks to make GANs work,2016,soumith
    https://github.com/soumith/ganhacks

11. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, 2017 https://arxiv.org/abs/1706.08500

12. Finding connections among images using CycleGAN, https:// www. youtube. com/watch?v=Fkqf3dS9Cqw&t=2799s, 2017년 10월

13. 1시간만에 GAN(Generative Adversarial Network) 완전 정복하기, https:// www. youtube.com/watch?v=odpjk7_tGY0&t=3329s, 2017년 8월

14. 10분안에 배우는 머신러닝 GAN 알고리즘 원리와 응용분야, https://www.youtube.com/watch?v=N9ewzLUZhL8, 2018년 12월

15. 실전! GAN 프로젝트, 카일라쉬 아히르와 지음, 위키북스, 2019년 11월

16. 실전 예제로 배우는 GAN, 조시 칼린 지음, 위키북스, 2019년 7월