iGAN을 활용한 '캐릭터 자판기' Project 최종발표

> 000조 고민주 오영진 오현지

INDEX

- 01 Project 목적
- 02 iGAN 모델 적용
- 03 SRGAN 모델 적용
- 04 Project 진행과정
- 05 Project 결과
- 06 Project 한계점

역할분담 및 Reference

01 Project 목적

게임 캐릭터 디자이너가 게임 속 특정 배경이나 컨셉이 필요한 캐릭터를 디자인할 때 원하는 이미지의 캐릭터 초안을 제공

- 캐릭터의 단순 생성이 아닌, 게임의 분위기와 스토리에 맞는 캐릭터 창작 필요
- 캐릭터 디자이너의 요구사항에 따른 다양한 캐릭터 생성이 필요



=> KT 에서도 게임 제작 시 캐릭터 디자인에 이용가능

[그림1] 엘도라도 게임이미지

01 Project 목적

특정 컨셉이 필요한 캐릭터를 디자인하는데 드는 어려움을 해소

하나의 컨셉으로도 다양한 캐릭터 생성 가능 [그림2] 캐릭터 이미지

기존 iGAN 논문 - "Generative Visual Manipulation on the Natural Image Manifold"

DCGAN을 이용하여 캐릭터 생성

+

HogNet을 이용하여 interactively 형태 변환

추가한 기술

- 1. 기존에는 그림을 그려 캐릭터를 생성하고 형태를 변환(HogNet 이용)하였지만, 특정 컨셉을 가지는 캐릭터를 생성하기 위하여 사진 파일을 첨부하면 Z-vector를 예측(AlexNet 이용)하여 캐릭터가 생성되도록 하였다.
- 2. 하나의 Z-vector를 생성한 후 noise를 추가하여 16개의 캐릭터가 생성되도록 하였다.

Generator와 Discriminator Model(DCGAN) 학습은 동일

1. Z-vector Predict Model 학습 (AlexNet)

```
파일 첨부하기 → 학습된 AlexNet 이용 → Inverse Generator가 Z-vector 예측
→ 학습된 Generator 이용 → 캐릭터 생성
```

2. Interactive Model 이용 (HogNet)

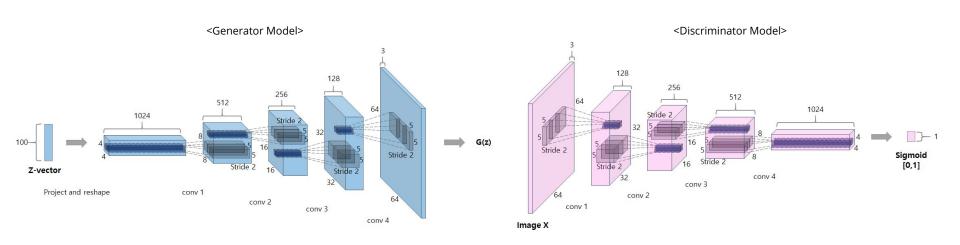
```
그림 그리기 → 학습된 HogNet 이용 → Z-vector 출력
→ 학습된 Generator 이용 → 캐릭터 생성
```

Model 학습 Parameter (DCGAN, AlexNet)

Model Constructing Input Parameter			Model Constructing Input Parameter		
Parameter	값	의미	Parameter	값	의미
n_layers	3	# of convolution layers	niter_decay	25	# of iter to linearly decay learning rate to zero
n_f	128	# of (last) feature channels	init_sz	4	# of first feature map size(4*4)
npx	64	height (= width)	fs	5	Fractionally-strided convolution filter size
nc	3	# of image channels (RGB)	bs	32	batch_size
nz	100	# of dim for Z-vector	lr	0.0002	learning_rate
niter	25	# of iter at starting learning rate	b1	0.5	Momentum

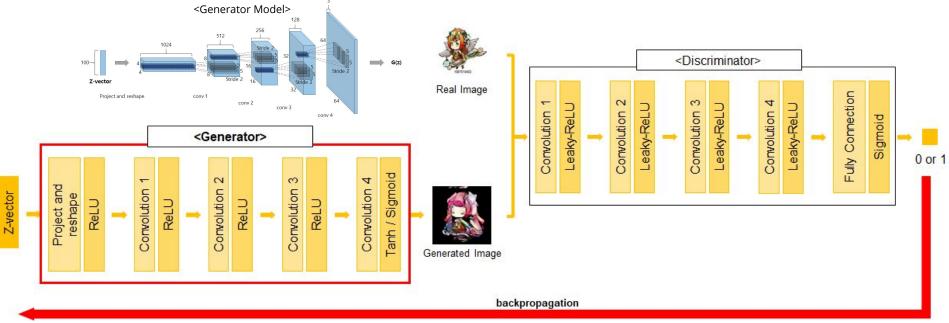
[丑1] Model Training Parameter

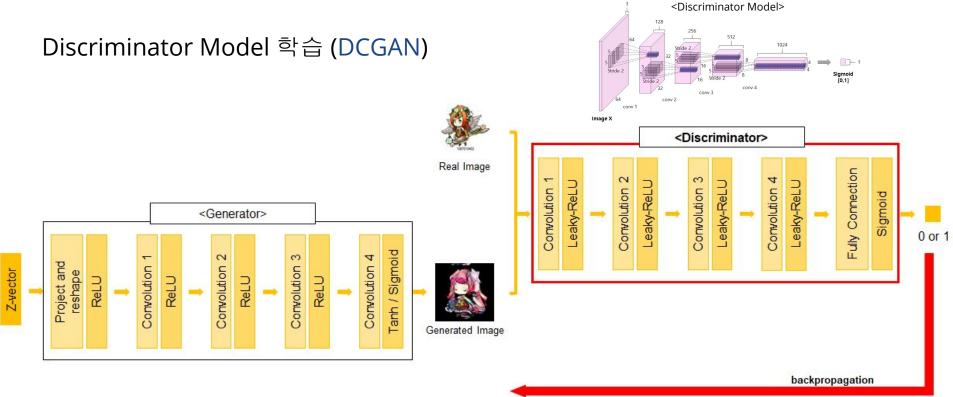
Generator와 Discriminator Model 학습 (DCGAN - minmax)



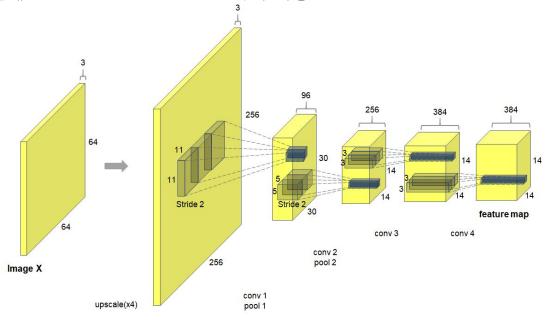
[그림3] DCGAN Model (Generator와 Discriminator Model)

Generator Model 학습 (DCGAN)

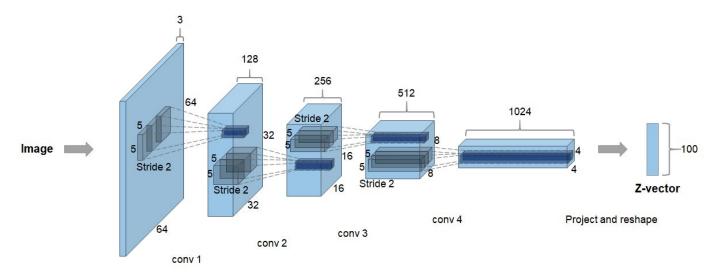




- Z-vector predict Model (AlexNet)
 - AlexNet의 4번째 Convolutional Network까지 이용

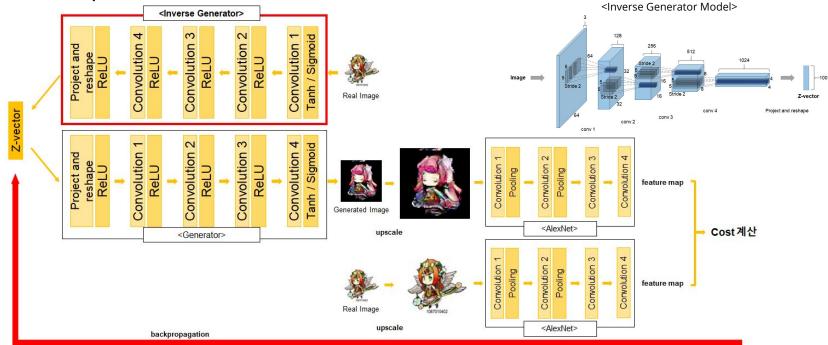


- 1. Z-vector predict Model (Inverse Generator Model)
 - Generator Model을 반대 방향으로 학습

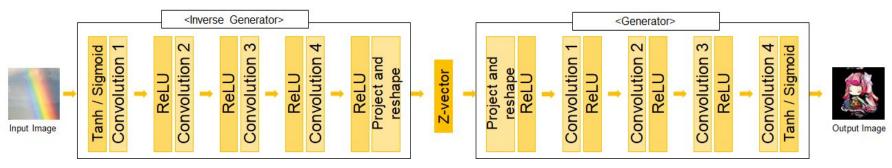


[그림7] Inverse Generator Model

1. Z-vector predict Model 학습 (Inverse Generator Model)

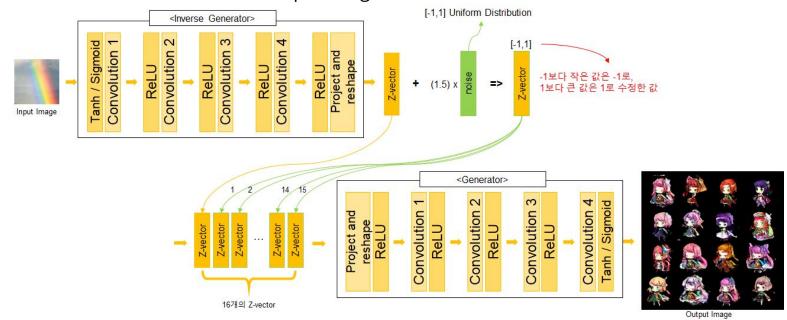


1. Z-vector predict Model 적용 결과

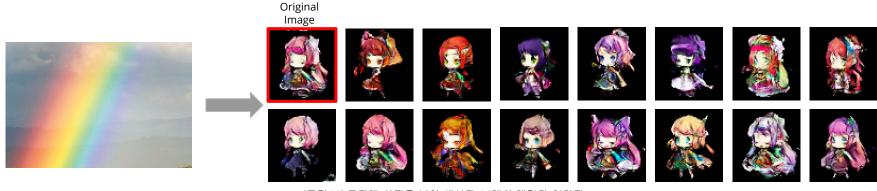


[그림9] 학습된 Inverse Generator와 Generator Model을 이용한 결과 : Input Image를 넣고 학습된 Inverse Generator Model을 이용하여 Z-vector 예측 후 학습된 Generator Model을 이용하여 개릭터 생성

- 1. Z-vector predict Model 적용 결과
 - Z-vector에 noise를 넣어 16개의 output image 출력

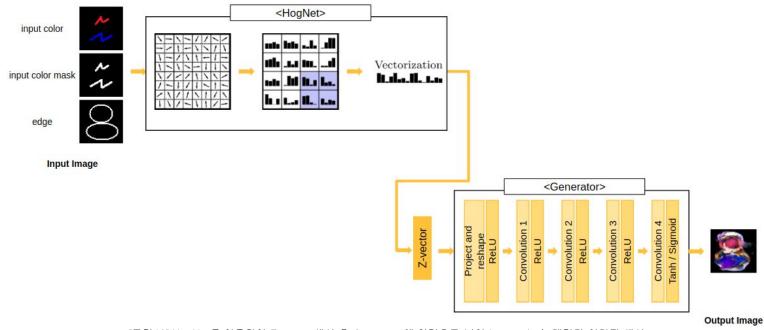


- Z-vector predict Model 적용 결과
 - Z-vector에 noise를 넣어 16개의 output image 출력

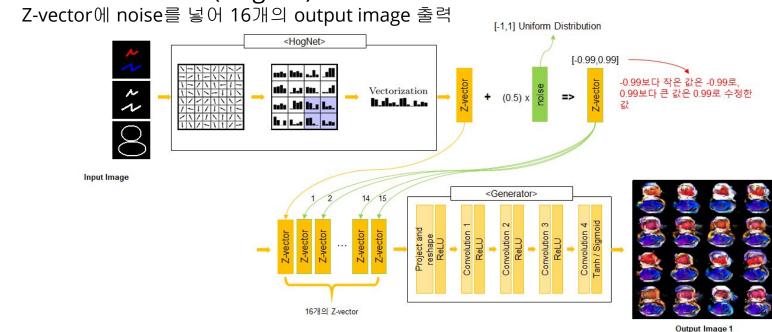


[그림11] 무지개 사진을 넣어 생성된 16개의 캐릭터 이미지

2. Interactive Model (HogNet) 적용 결과



2. Interactive Model (HogNet) 적용 결과



iGAN의 한계점

- 64x64 이미지의 경우만 좋은 성능을 보여주어 낮은 해상도의 캐릭터가 생성



[그림14] 생성된 캐릭터 이미지 결과

=> 해상도를 높여주는 Super-Resolution GAN 모델을 이용

SRGAN 논문 - "Image Super-Resolution Using Deep Convolutional Networks"

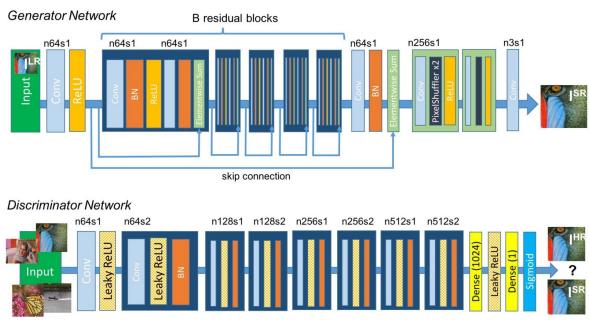
DCGAN을 이용하여 저해상도 이미지를 고해상도 이미지로 변환하는 기법

SRGAN 모델을 이용하여 선명한 캐릭터 생성



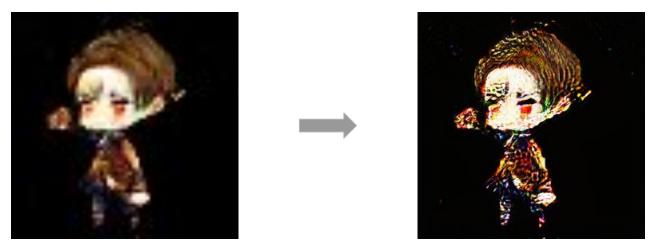
[그림15] 생성된 저해상도 캐릭터 이미지를 이미 학습된 SRGAN Model을 이용하여 고해상도 캐릭터 이미지로 변환한 결과

SRGAN Model (DCGAN)



기존 논문에서 학습된 SRGAN Model 이용한 결과

- 해상도는 발전되었지만 깨지는 현상



[그림17] 생성된 저해상도 캐릭터 이미지를 이미 학습된 SRGAN Model을 이용하여 고해상도 캐릭터 이미지로 변환한 결과

고해상도 캐릭터로 SRGAN Model 재학습

- 기존 논문에서 학습된 SRGAN Model을 초기값으로 설정하여 고해상도 캐릭터로 재학습
- 학습 parameter

Model Constructing Input Parameter				
Parameter	값			
batch_size	16			
learning_rate	1e-4			
momentum	0.9			
epoch	100			
decay_every	50			
Ir_decay	0.1			

[丑2] SRGAN Model Training Parameter

재학습된 SRGAN Model 적용 결과

- 자연스럽게 고해상도로 변환된 결과



[그림18] 생성된 저해상도 캐릭터 이미지(a)를 이미 학습된 SRGAN Model을 이용하여 고해상도 캐릭터 이미지로 변환한 결과(b)와 고해상도 캐릭터로 재학습된 SRGAN Model을 이용하여 고해상도 캐릭터 이미지로 변환한 결과(c)

04 Project 진행과정 - 개요

프로젝트명	캐릭터 자판기				
프로젝트요약	사용자가 게임 캐릭터를 디자인할 때 드는 어려움을 덜어주고자 하는 프로젝트로, 사용자가 원하는 이미지의 특정 배경을 선택하거나스케치하면 그러한 특징을 갖는 게임 캐릭터가 생성된다.				
팀명 GOO		팀원	고민주 오영진 오현지		
대분류	대분류 영상		Image Generation		
AI 모델	DCGAN	학습방식	Unsupervised Learn	ing	
	Docker 환경구성	Ubuntu 16.04, CUDA 8.0, cuDNN 8.0.61			
AI 개발환경	Docker Hub주소	https://hub.docker.com/r/ggmmjj1/mini4/ https://hub.docker.com/r/oyj9097/srgan/			
אואופנט	프레임워크(버전)	Tensorflow 1.4			
	언어(버전)	Python3			
	기타(IDE 등)	spyder3, openCV3, theano,	pyqt4, hdf5	t4, hdf5	
코드 및 데이터 파일	Google 드라이브 주소	https://drive.google.com/open?id=17CEvcdjjOP3jz5p2hFiT_ecbJTVcyUUw (코드) https://drive.google.com/open?id=1Z-6c61dPqspxCrs7vQ4ZIDIdqGa47i9- (데이터)			
데모 스펙	HW(장비포함)	운영체제: Linux ktai03-Alienware-Aurora-R7 4.13.0-43-generic #48~16.04.1-Ub SMP Thu May 17 12:56:46 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux 프로세서: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz 메모리: 31GiB		eneric #48~16.04.1-Ubuntu	
	SW	GUI: pyqt4			
	데모영상 Google 드라이브 주소	https://drive.google.com/o	pen?id=1a5JWn8s8nqNrFxZsQ	mrnCtUxhD7Gxmvd	

04 Project 진행과정 - 개요

데이터셋



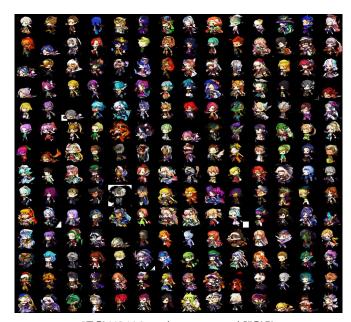
데이터셋	저해상도 캐릭터	고해상도캐릭터	
출처	http://yurudora.com/tkool/		
개수	1023x64	40,000개 (1,034개를 이미지 변환함)	
데이터 구성	64 x 64 게임 캐릭터	128 x 128 게임 캐릭터	

1. 게임 캐릭터 dataset 학습 (오영진)

- 약 6만개의 캐릭터 이미지 (64x64)
- 학습 parameter(slide 7 참고)로 Model 학습
- 코드수정 (DCGAN) 관련 코드 파일(train_dcgan): [train_dcgan.py, train_dcgan_config.py]
 - charac_64.hdf5 dataset
 - 학습 Parameter(slide 7 참고) 변경: epoch → 100

(Inverse Generator) 관련 코드 파일: [train_predict_z.py]

- charac_64.hdf5 dataset
- pretrained Alexnet
- 학습 Parameter(slide 7 참고) 변경: epoch → 100



[그림19] 196 random generated 캐릭터

2. 기능 추가 (오현지)

- 1개의 Input으로 16개 캐릭터를 생성하여 다양성 제공하고자 함
- Z-vector에 15개의 noise를 추가하여 16개의 캐릭터 생성
 - Z-vector Predict Model에 의해 생성된 Z-vector에 15개의 랜덤 noise-vector추가 (noise-vector는 [-1,1] uniform distribution을 따르는 길이 100인 vector)
 - 생성된 Z-vector + 1.5 x noise-vector
 - Z-vector가 [-1,1] 범위가 되도록 -1 보다 작은 값은 -1로, 1 보다 큰 값은 1로 수정
 - 생성된 16개 Z-vector = 생성된 1개의 Z-vector + noise를 추가한 15개의 Z-vector
 - 학습된 Generator를 이용하여 16개의 캐릭터 이미지 생성
- 코드수정
 - 관련 코드 파일(iGAN-master): [iGAN_predict.py]
 - def.predict_z2 추가 : noise vector 생성 후 기존 Z-vector에 1.5를 곱하여 noise 추가된 Z-vector 생성
 - def.invert_image_CNN_opt2 추가 : noise가 추가된 Z-vector를 이용하여 optimize
 - def.parse_arg() 수정 : RGB 형태의 이미지가 아닌 경우 RGB 형태의 이미지로 변환, jpg or png 파일에 따라 결과 나오도록 수정, num_z(noise-vector의 개수)=15, 생성된 캐릭터 이미지 저장(파일 경로, 이름 지정)

3. UI 구현 및 기능 수정 (고민주)

- pyqt를 이용해 UI 레이아웃에 대한 디자인을 구현
- 메뉴얼 다이얼로그
 - 레이아웃 상단에는 간단한 설명이 나오고, 아래 버튼을 클릭하면, 설명서에 대한 상세 이미지들이 gif파일로 QDialog위젯 상에 나옴.
- 이미지 파일 첨부해서 추가하기
 - 파일 추가 버튼을 클릭시, 왼쪽 화면에서 draw할 수 있는 위젯을 파일 이미지로 보여주기 위해 stackedWidget을 이용
 - drawWidget과 이미지 파일을 보여주는 label widget 간에 stack index를 switch하는 방식으로 처리
 - subprocess함수를 통해 main함수가 아닌 predict.py shell script를 호출해서 첨부한 이미지에 대한 캐릭터를 생성하도록 함.
- 코드수정 관련 코드 파일(iGAN-master): [gui_design.py]

3. UI 구현 및 기능 수정 (고민주)

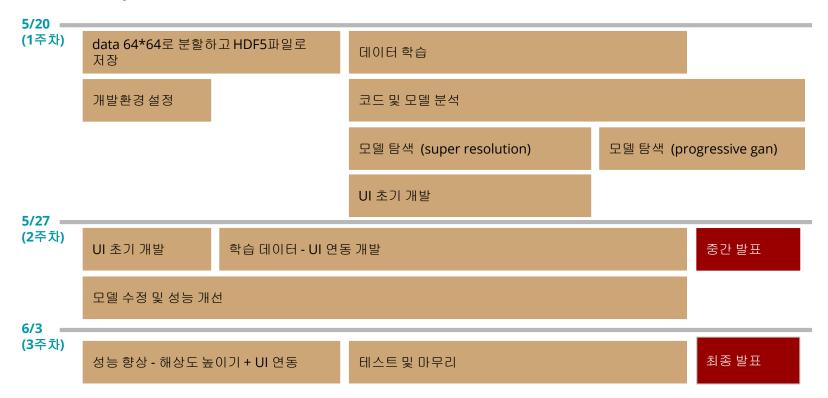
- Interactive Generator에 고해상도 이미지 저장 기능 추가
 - save버튼을 누르고 디렉토리를 설정하면 고해상도 이미지로 저장
 - Interactive Generator는 고해상도를 적용하면 Interactive한 기능이 사라짐
- Network based Z-vector predict Generator로 생성된 16가지 캐릭터 이미지 고해상도로 불러오기 기능 추가
 - File버튼을 누르고 컨셉 이미지를 입력하면 16개의 캐릭터 이미지를 저장
 - 만들어진 이미지를 고해상도로 변경해서 화면에 바로 불러오기 적용
 - predict.py를 통해 생성된 16개의 이미지는 오른쪽 화면에서 gridLayout위젯에서 4x4형식으로 나타나게함.
- 코드수정 관련 코드 파일(iGAN-master): [iGAN_main.py]

3. UI 구현 및 기능 수정 (고민주)

- 로딩 화면
 - 이미지 파일을 넣을때 로딩화면에 화면에 중첩되어 나타나게 구현한 코드이다. QTimer를 이용해 5초 가량 indicator들이 애니메이션 효과로 나타났다가 사라지게 된다.
- 코드수정 관련 코드 파일(iGAN-master): [gui_loading.py]
- Splash 화면
 - QSplashScreen 위젯을 통해 프로그램 시작 시 4초 간 splash 화면 실행
- 코드수정 관련 코드 파일(iGAN-master): [iGAN_main.py]

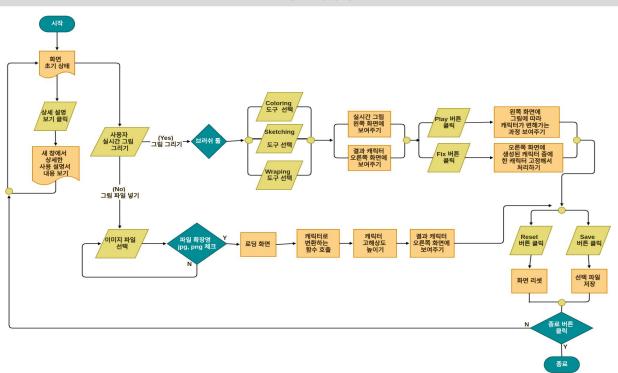
4. SRGAN 기능 수정 (오영진)

- SRGAN 학습
 - 약 1023개의 캐릭터 이미지 HR(400x400), LR(96x96)
- IGAN에서 이미지를 만들 시점에서 고해상도로 변경해주는 코드를 실행하도록 변경
 - main.py파일의 evaluate 함수를 반복해서 실행하여 이미지 생성을 감지하면 고해상도 기능을 실행하도록 변경
 - IGAN과 SRGAN이 workspace를 적절하게 공유하면서 고해상도 기능을 수행
- 코드수정 관련 코드 파일(sr-master): [main.py --mode=evaluate]



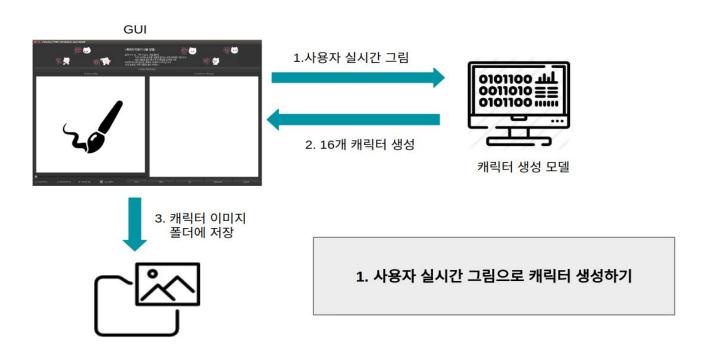
05 Project 결과

데모 시나리오



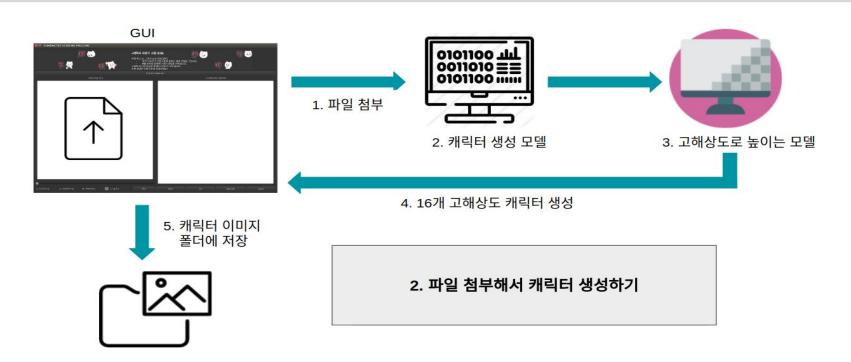
05 Project 결과

프로젝트 구조도



05 Project 결과

프로젝트 구조도



BEST CASE

(1) 이미지의 사진이 비슷한 단일 색상으로 이루어진 경우



BEST CASE

(1) 이미지의 사진이 비슷한 단일 색상으로 이루어진 경우



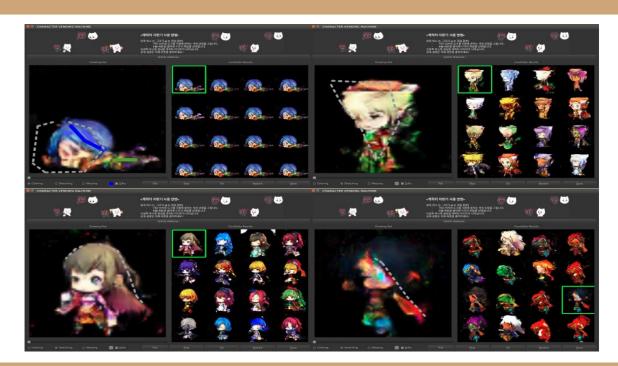
BEST CASE

(2) Coloring 색상이 제한적인 경우



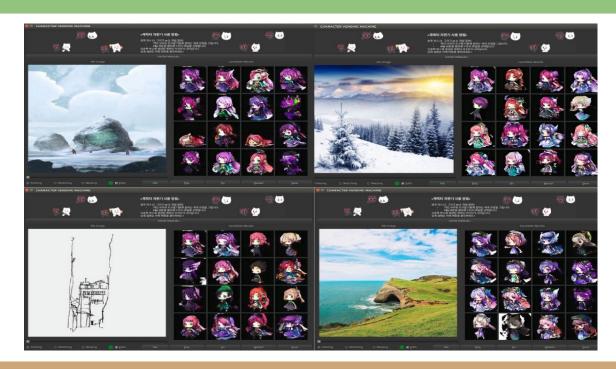
BEST CASE

(3) 캐릭터에 자주 나오는 Sketch 선을 그린 경우



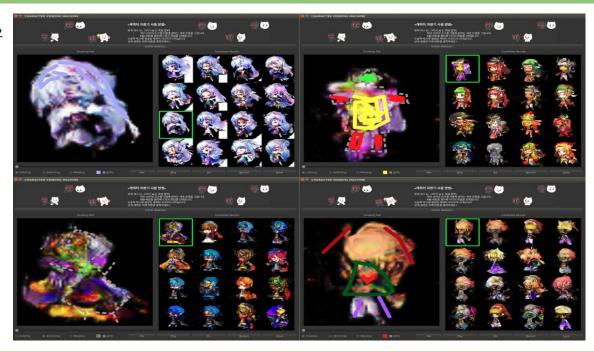
WORST CASE

(1) 이미지에 하얀 색이 많이 들어간 경우



WORST CASE

- (2) coloring 색상이 많은 경우
- (3) sketching에서 캐릭터에 없는 모양을 그린 경우



BEST CASE

WORST CASE





06 Project 한계점

DCGAN 자체 해상도 높이기

- 고해상도 캐릭터 dataset(1,034개)를 128x128로 resize하여 DCGAN(4 layers)로 변경하여 학습
- Image Augmentation을 통해 40,000개의 데이터 추출하여 학습
 - batch size => 64 => 32 => 128
 - momentum 0.5/0.8
 - z-vector 100 -> 200 ->400
 - Discriminator layer 축소
- 좋지 않은 학습 결과



[그림20] DCGAN을 이용하여 생성된 학습 이미지 결과

06 Project 한계점

Progressive GAN

- 1024x1024 사이즈까지 이미지를 생성하는 모델
- 16G memory x 8개 multi-gpu로 학습시 약 2일의 학습시간이 걸림
- 현재 사용가능한 하드웨어 성능으로 학습시킬 수 있도록 코드를 수정하는데 시간상 제약이 있음

역할분담

고민주

- 논문 분석
- UI 코드 분석/수정
- UI와 학습 모델 연동
- Test 및 디버깅
- 프로그램 Architecture
- 결과 케이스 분석

오영진

- 논문 분석
- Data 전처리
- DCGAN모델 학습
- SRGAN모델 학습

오현지

- 논문 분석
- iGAN 코드 분석/수정
- hyperparameter 정리
- 학습 모델 Architecture

Reference

iGAN

-(논문) Jun-Yan et al. "Generative Visual Manipulation on the Natural Image Manifold", in European Conference on Computer Vision (ECCV). 2016. (https://arxiv.org/pdf/1609.03552.pdf)

-(코드) https://github.com/junyanz/iGAN

-(Youtube 영상)https://www.youtube.com/watch?v=9c4z6YsBGQ0

SR-GAN

-(논문) Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." arXiv preprint(2017).

-(코드) https://github.com/tensorlayer/srgan

DC-GAN

-(논문) RADFORD, Alec; METZ, Luke; CHINTALA, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.(https://arxiv.org/abs/1511.064343)

-(코드) https://github.com/SeitaroShinagawa/DCGAN-chainer

Progressive-GAN

-(논문) KARRAS, Tero, et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. (https://arxiv.org/abs/1710.10196)

-(코드) https://github.com/tkarras/progressive_growing_of_gans

Data

- -http://yurudora.com/tkool/
- -http://machinelearninguru.com/deep_learning/data_preparation/hdf5/hdf5.html [HDF5 file 만들고 사용하기]

Reference

HOG(Histograms of Oriented Gradients)

-(이론) https://blog.naver.com/jh_h1022/220861693757

-(논문) https://lear.inrialpes.fr/pubs/2005/DT05/hog_cvpr2005.pdf

UI

- -https://wiki.python.org/moin/PyQt/A%20full%20widget%20waiting%20indicator
- -https://www.tutorialspoint.com/pyqt/pyqt_layout_management.htm
- -http://pyqt.sourceforge.net/Docs/PyQt4/qstackedwidget.html
- -http://nullege.com/codes/search/PyQt4

