

Pixir

원하는 스타일로 그려드립니다!

Text to Image

이성인 이현승 조지원 최신욱 홍세원

P X I R
Picture Shuttle



Sungin Lee



Hyeonseung Lee



Jiwon Cho



Sinwook Choi



Sewon Hong

INDEX

01 문제 인식

02 프로젝트 구성

03 프로그램 구조

04 결과

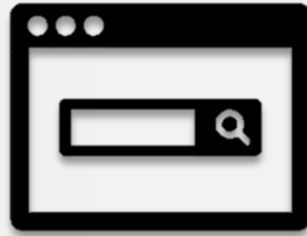
05 한계점

01

문 제 인 식



창작욕구



정보검색에 대한 시간절약



저작권 문제 해결

02

프로젝트 구성

프로젝트 개요

프로젝트 기간

2020년 01월 29일 ~ 2020년 03월 03일

데이터셋

1. KorQuad 2.0 dataset
2. MS-COCO dataset

사용 모델

Soyspacing, BERT, StackGAN, WGAN-GP, Neural Style Transfer

데이터셋

1. KorQuAD DATA(<https://korquad.github.io/>)

- 47,957개의 Wikipedia article에 대해 102,960개의 질의응답 쌍으로 이루어진 데이터셋
- 문단 뿐만 아니라 표와 리스트도 포함되어 있기 때문에 HTML tag 사용
 - ➡ 웹 문서, 약관, 표 등과 같은 다양한 길이와 형식을 가진 데이터셋을 이용하여 **한국어 띄어쓰기** 학습

2. MS-COCO DATA(http://www.aihub.or.kr/keti_data_board/visual_intelligence)

- 대규모 객체 감지, 분할, 캡셔닝 데이터셋
- 약 20만장의 이미지에서 전체 80종류 150만개 객체들에 대한 영역 분할(instance segmentation) 정보 포함
 - ➡ MS-COCO의 영어 캡션을 **한국어로 번역**된 데이터셋을 이용하여 학습

03

프로그램 구조

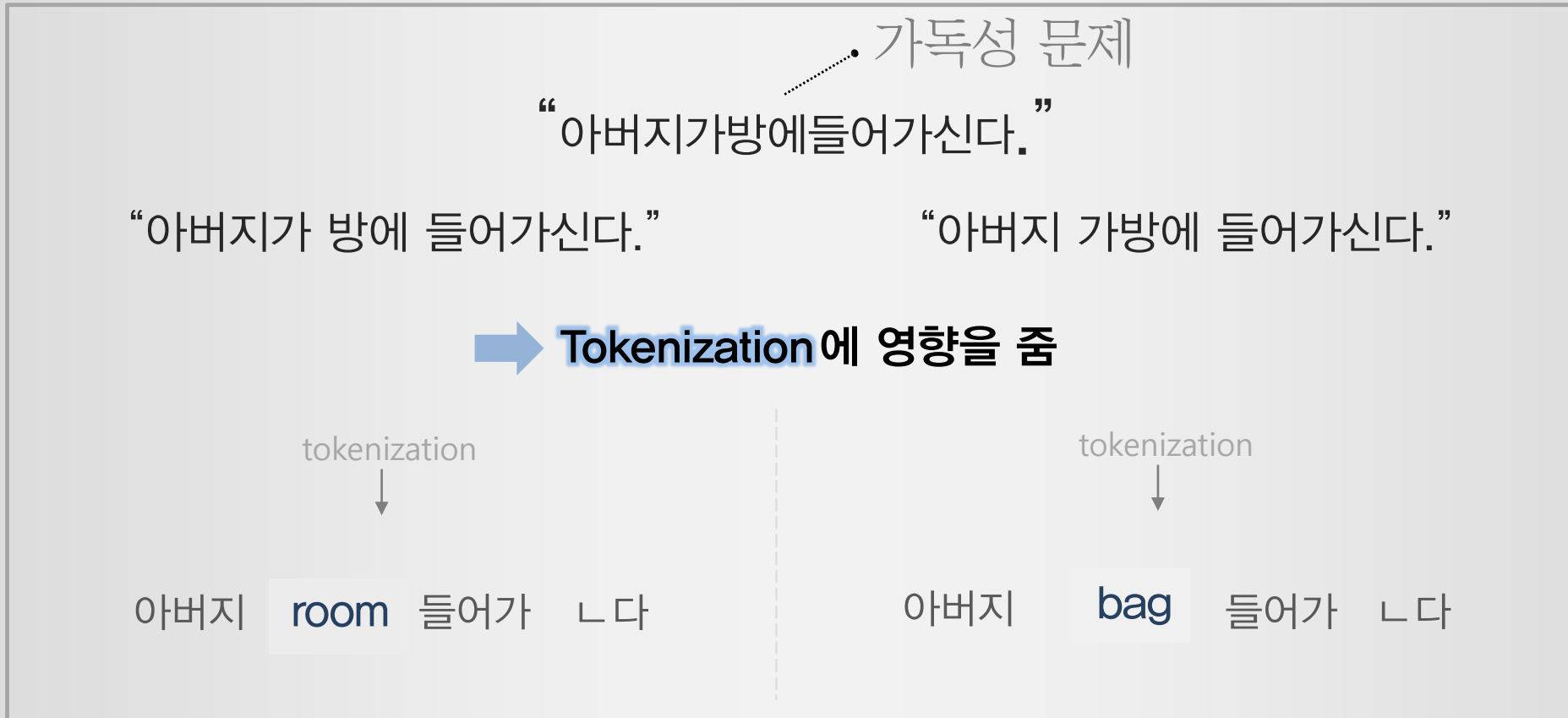
“공원에서저글링을하는남자”



“공원에서 저글링을 하는 남자” → [0.0125 0.3421 0.1242 ...]





Soyspacing



Soyspacing

- 102,960개의 KorQuAD data를 학습시킴
- KorQuAD data를 통해 학습이 되지 않은 문장에 대해 띄어쓰기 규칙 적용

>> Korean space corrector 객체 생성

“개가저글링을한다”  “개가 저글링을 한다”  띄어쓰기 규칙 적용

개가	→	0
저글링을	→	1
한다	→	2

BERT(Bidirectional Encoder Representation Transformer)

“아버지가 방에 들어가신다.”



아버지 방 들어간다



[0, 1, 2]

“아버지가 방을 리모델링 하신다.”



아버지 방 리모델링한다



[0, 1, 3]

“아버지가 방에서 나오신다.”



아버지 방 나온다



[0, 1, 4]

한국어분석 태스크
(의미역인식 성능 : 85.77%)

기계 독해 태스크
(KorQuAD dev셋 : 94.18%)

문장/문서분류 태스크
(단락 재순위화 : 73.73%)

언어모델 단위 별 문맥 반영 워드 벡터

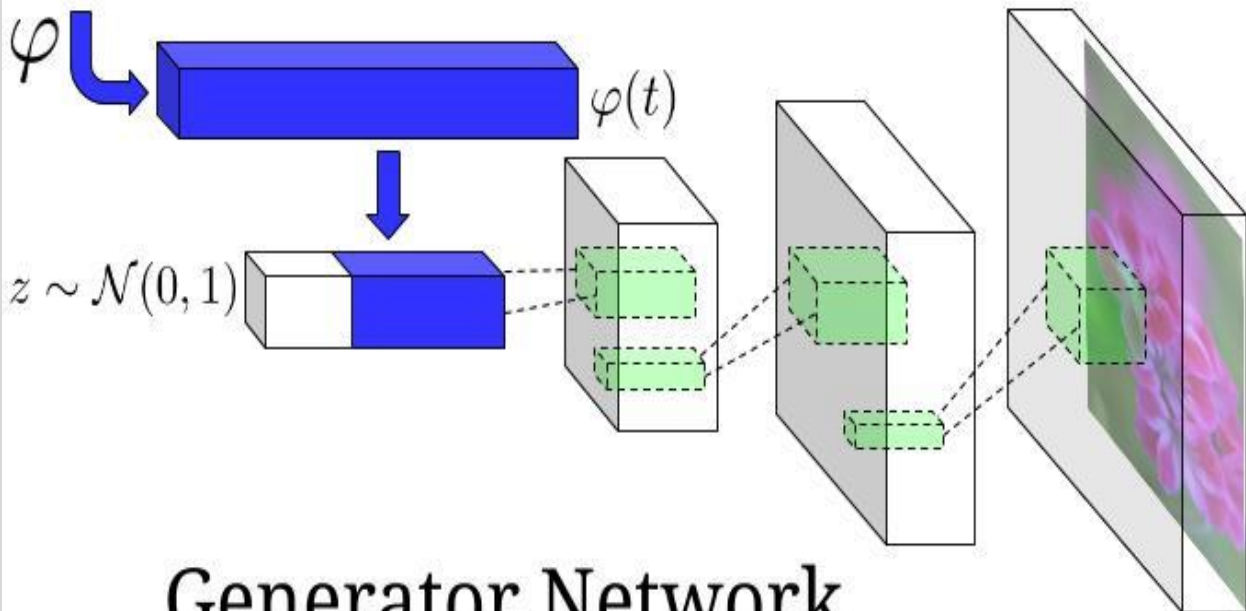


GAN

세부적이고 시각적인 묘사를 바탕으로 이미지를 생성하는 간결하고 효과적인 모델

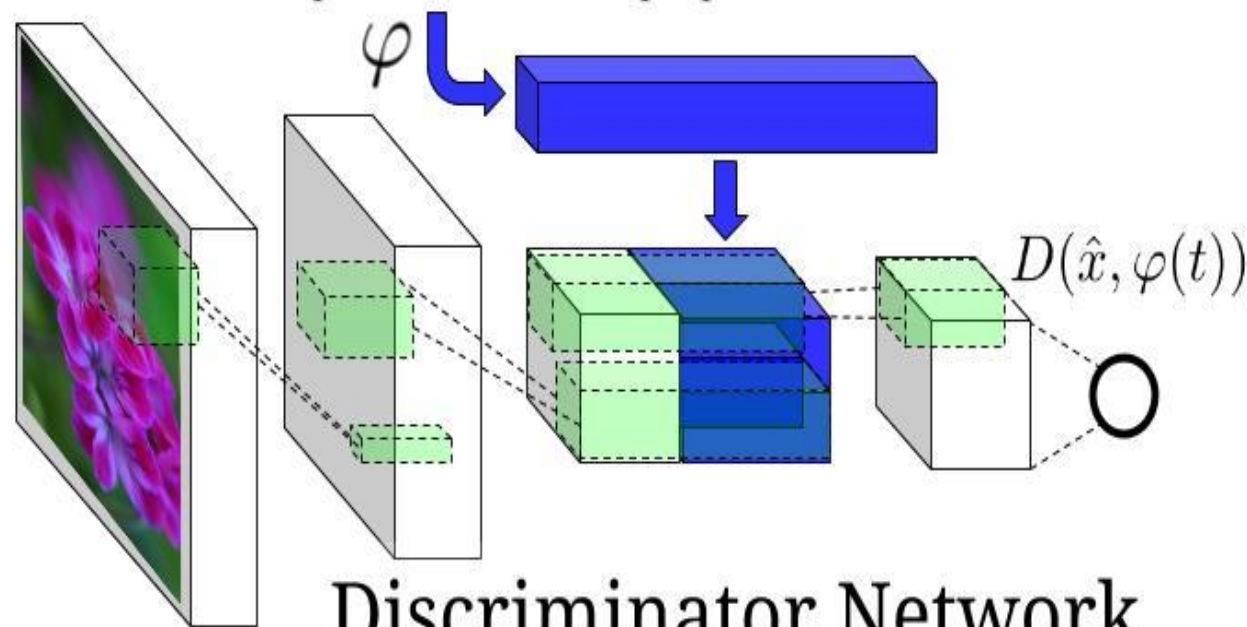
Manifold interpolation regularizer로 text to image의 성능을 향상

This flower has small, round violet petals with a dark purple center



Generator Network

This flower has small, round violet petals with a dark purple center

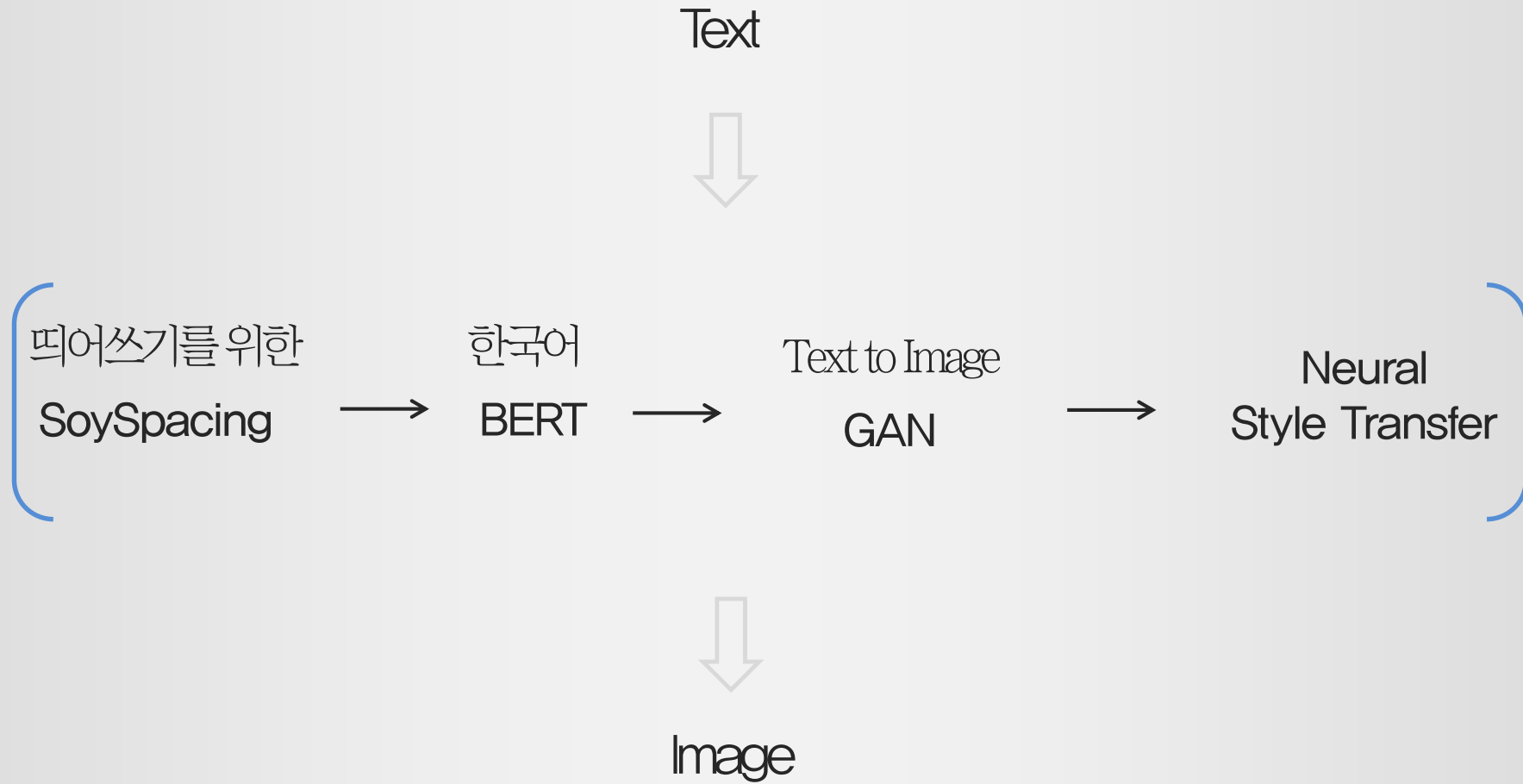


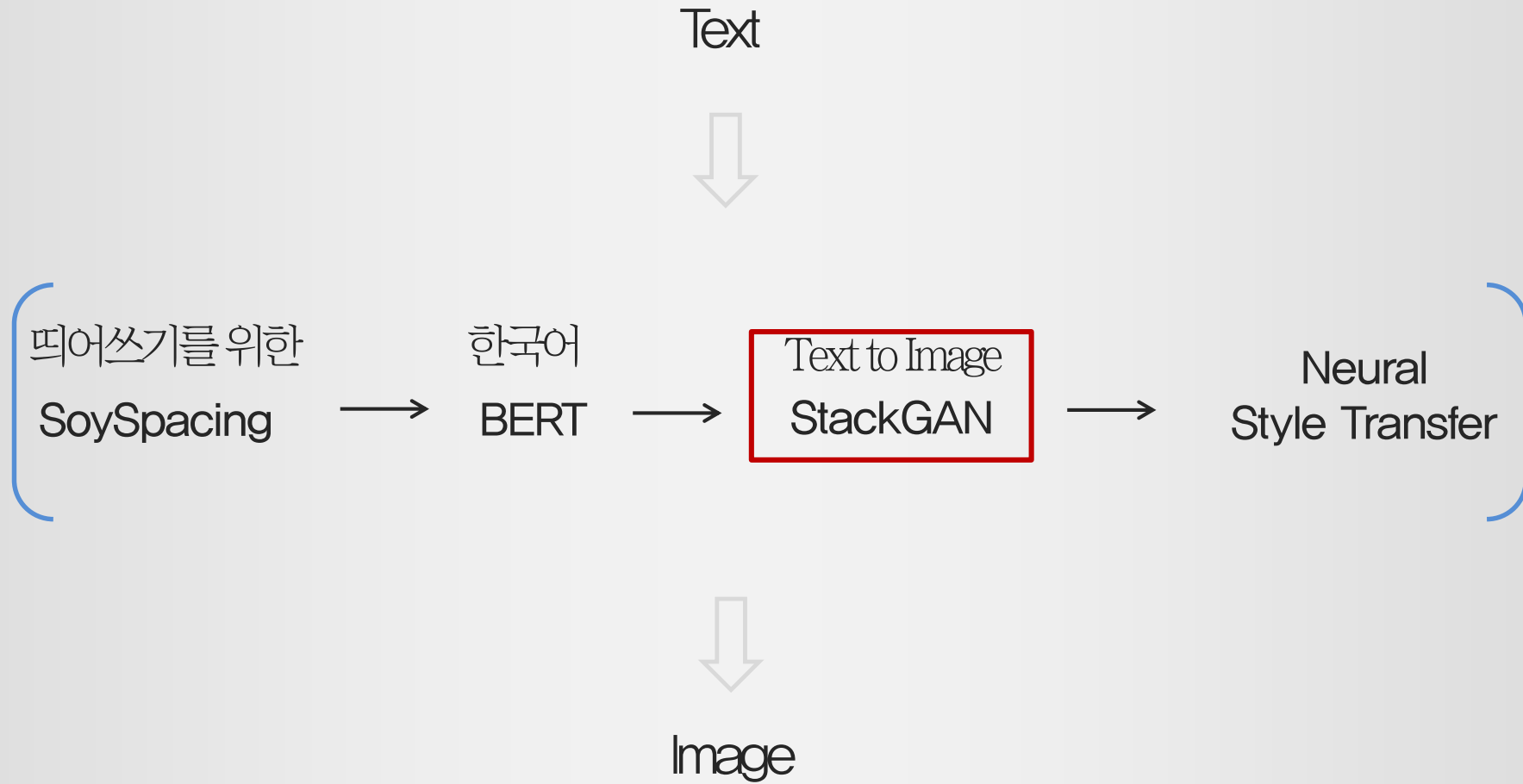
Discriminator Network

GAN

GAN을 사용해 생성한 이미지

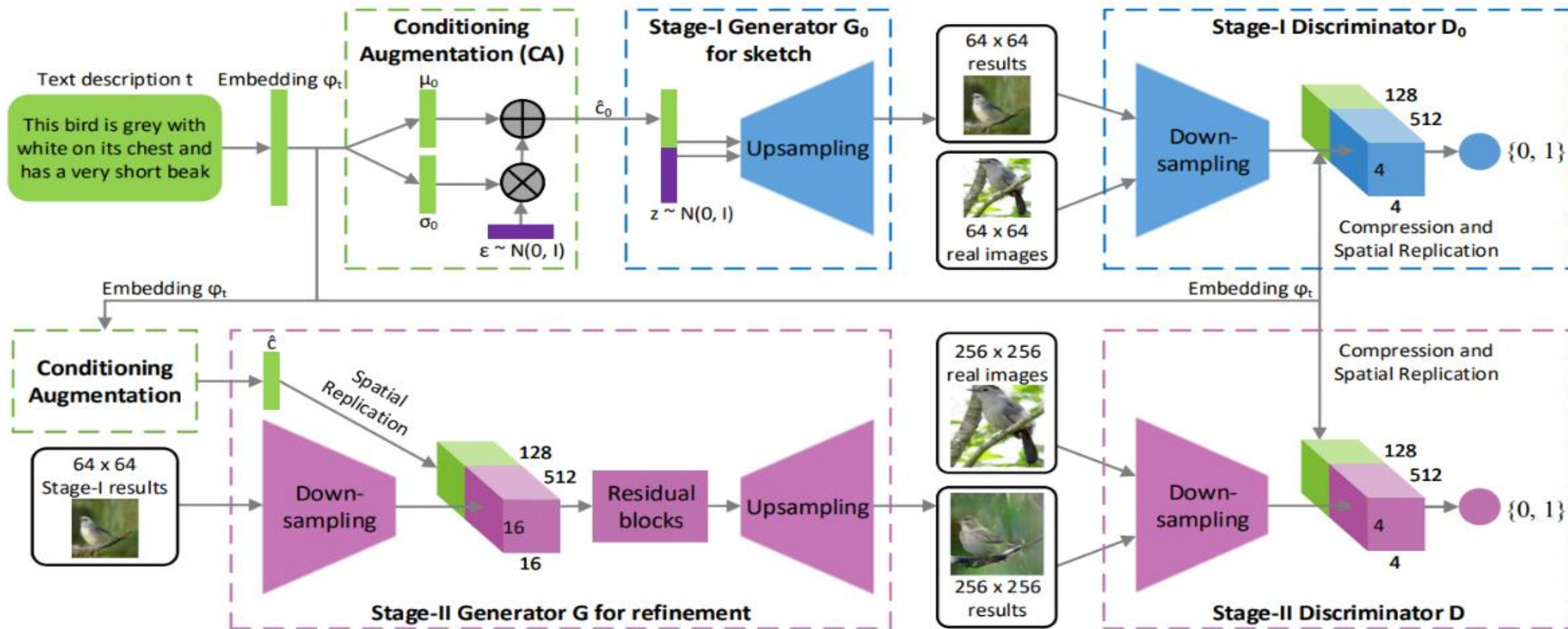


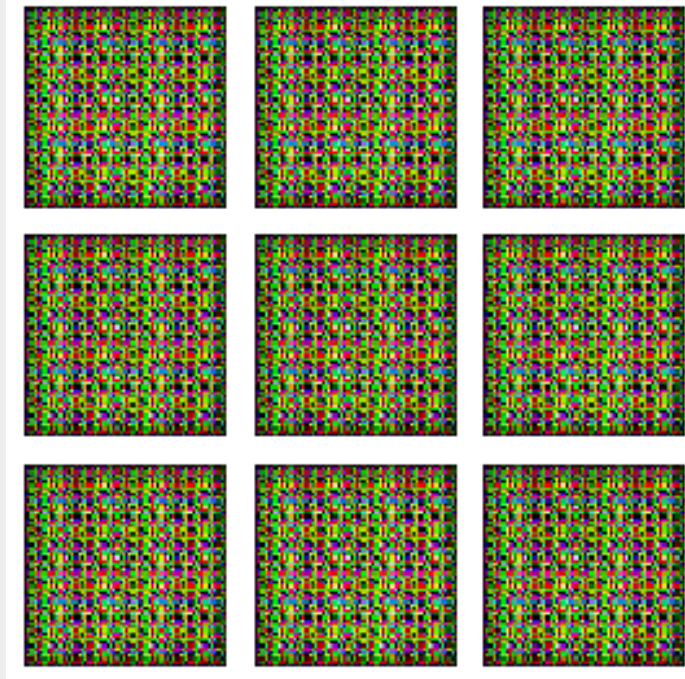




StackGAN

글로 묘사된 내용을 바탕으로 사실적인 이미지를 생성하는 신경망
겹쳐 쌓인(stacked) 두 개의 GAN이 모여 고해상도 이미지를 생성



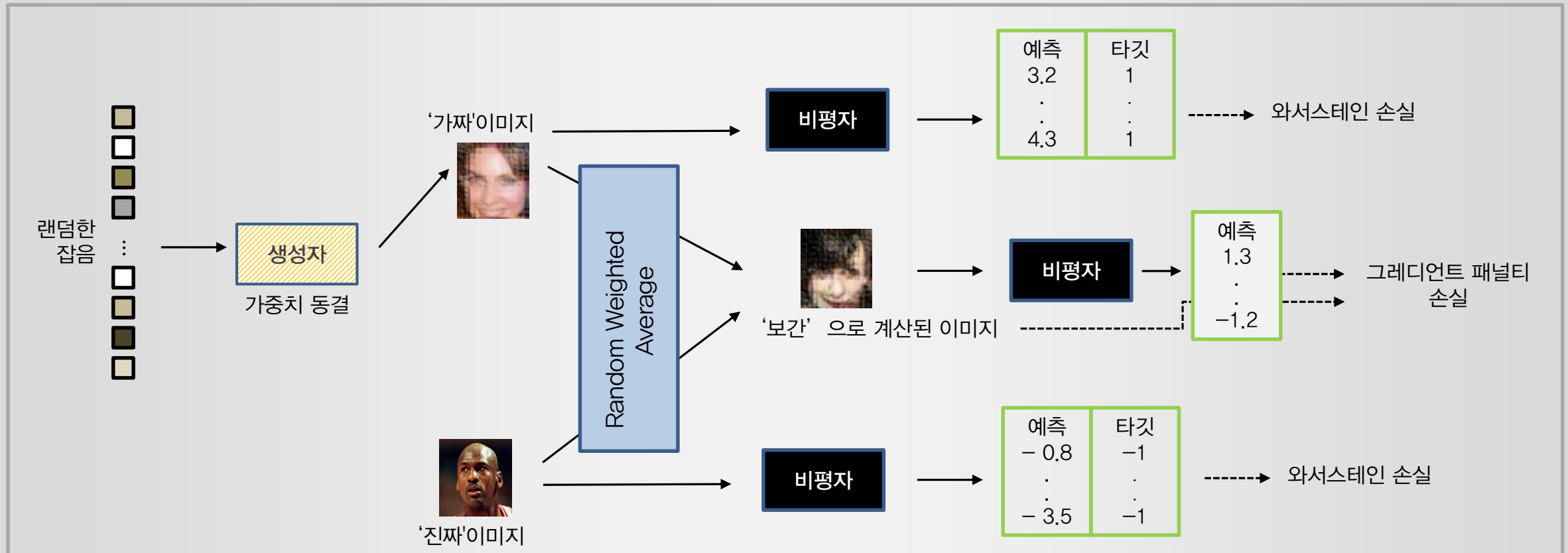






























➡ 모드붕괴 : 판별자를 반드시 속일 수 있는 임의의 값으로 수렴하는 현상

WGAN - GP(Wasserstein GAN - Gradient Penalty)

- ❖ WGAN : 판별자와 생성자를 위해 손실함수인 wasserstein 손실함수 사용
 - 생성자를 한번 업데이트할 때 판별자를 다섯 번 업데이트
- WGAN-GP는 WGAN과 생성자는 동일하지만 **판별자가 다름**
 - 판별자의 가중치를 클리핑하지 않고, 배치 정규화 층을 사용하지 않음
 - 판별자의 손실 함수에 **gradient penalty** 항 추가

WGAN - GP 비평가 훈련 과정



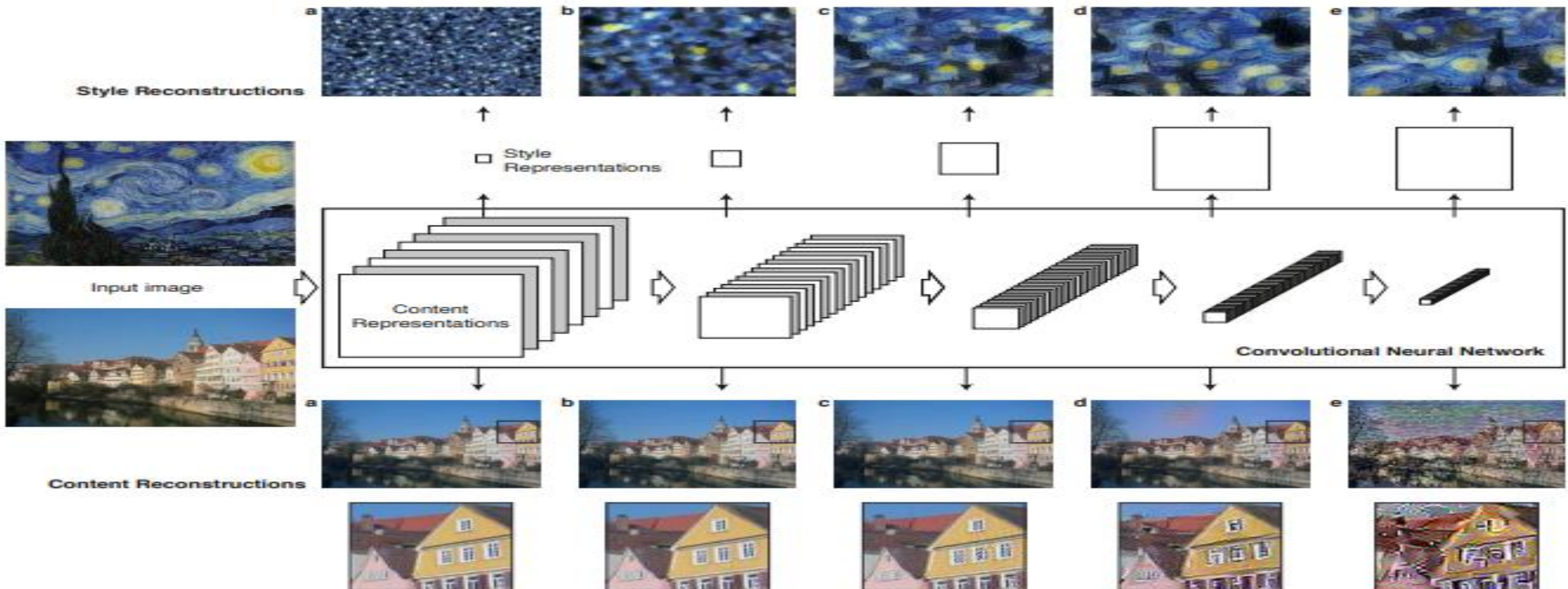
DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)
Baseline (G : DCGAN, D : DCGAN)			
			
G : No BN and a constant number of filters, D : DCGAN			
			
G : 4-layer 512-dim ReLU MLP, D : DCGAN			
			
No normalization in either G or D			
			
Gated multiplicative nonlinearities everywhere in G and D			
			
\tanh nonlinearities everywhere in G and D			
			
101-layer ResNet G and D			
			

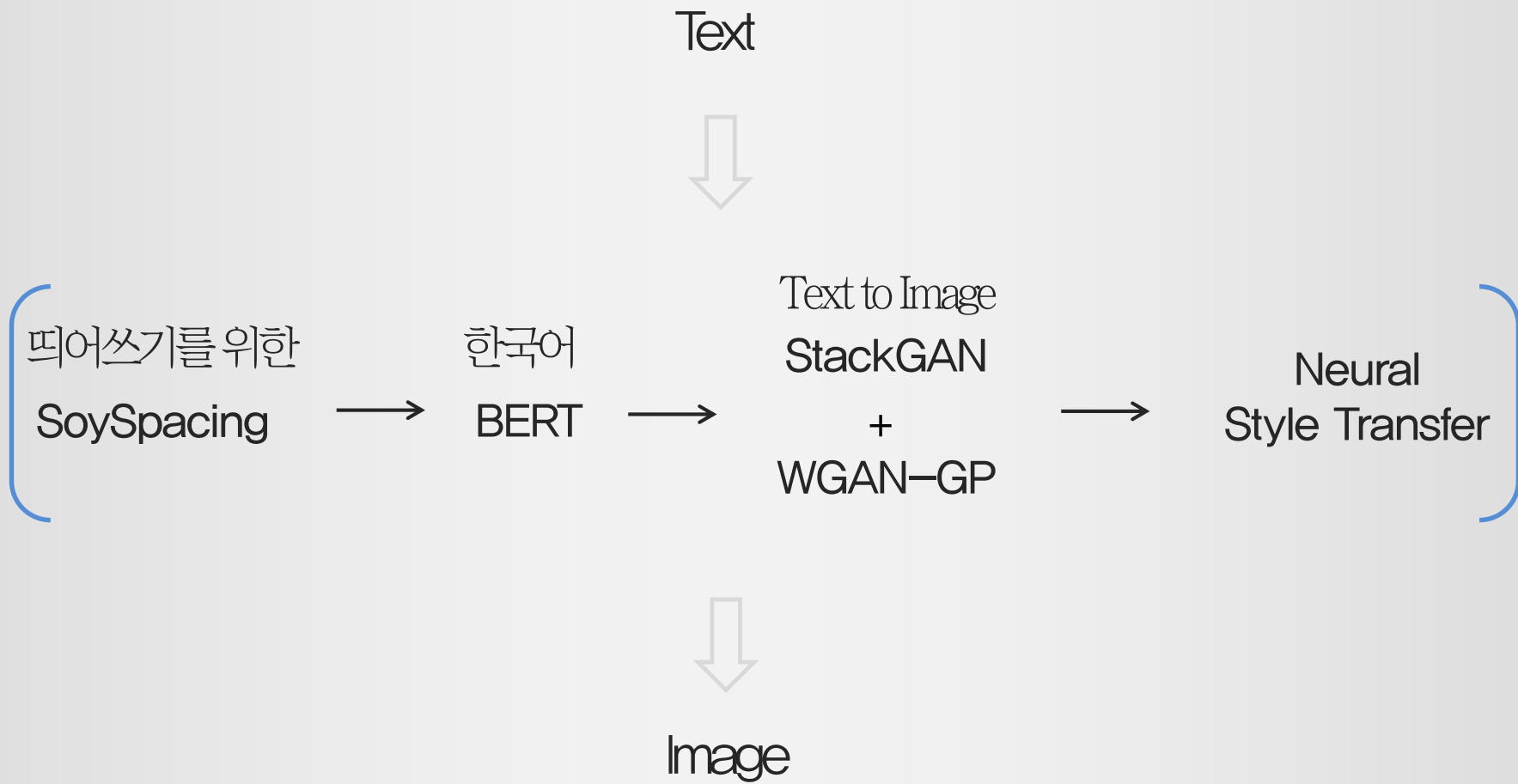
» GAN중 Image 성능 ↑

Neural Style Transfer

훈련세트를 사용하지 않고 이미지의 스타일을 다른 이미지로 전달해 변환

이미지의 내용을 식별하도록 성공적으로 사전 훈련된 VGG19 사용





04

결과

Text

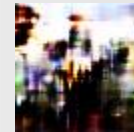
“개가 저글링을 하고 있다.”

“사람이 오토바이를 타고 있다.”

“공원에서 산책하는 사람.”



Image

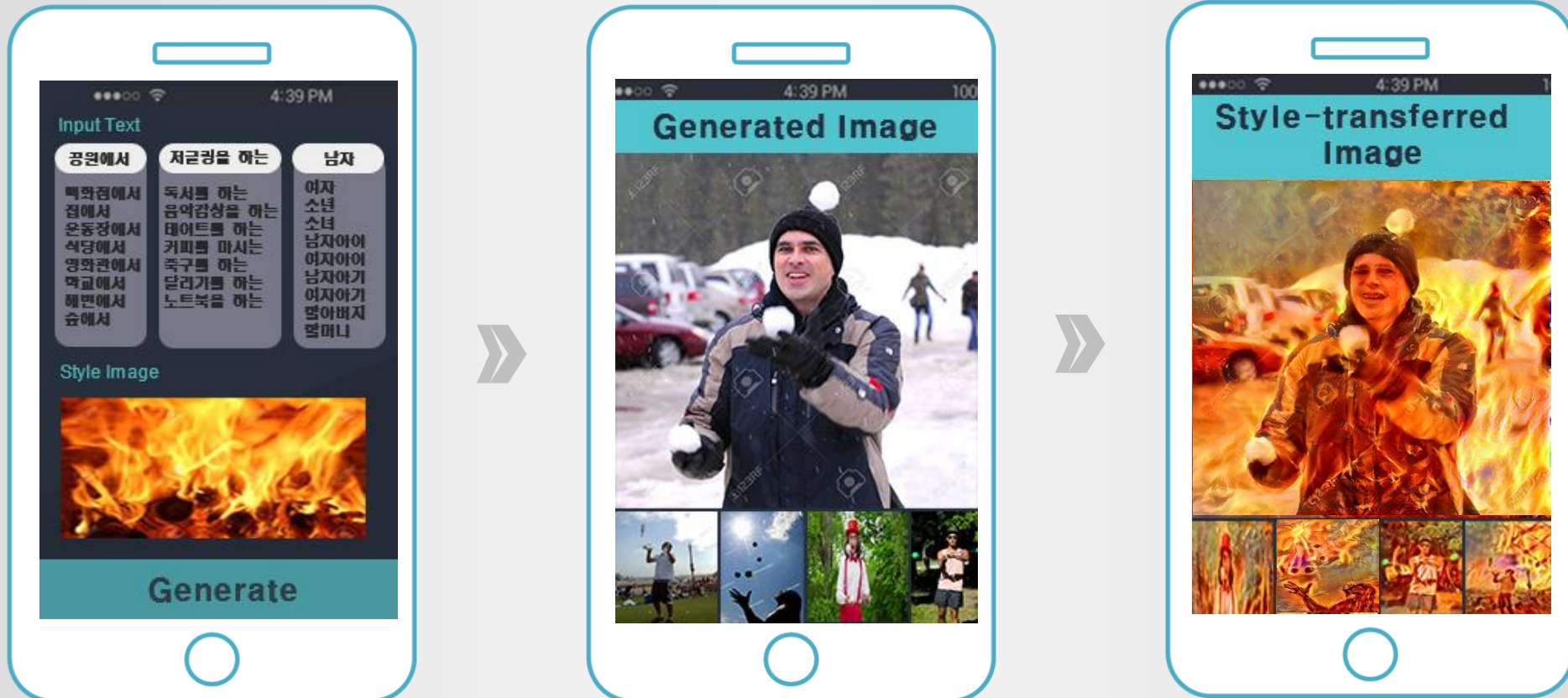


05

한계점

01 Tensorflow Lite와 하드웨어의 한계점

- 우리 수준에서 Tensorflow Lite로 변환하기 위해 Model을 작은 부분으로 나누고, 통합하는 class를 만들어서 어플리케이션에 구동하는 것이 어려움
- Tensorflow Lite도 불완전한 개발단계에 있어 지원하지 않는 명령어도 많고, 변환에 제약사항이 많음



02 시간적 한계점

Stage1과 Stage2 모두 구현하였으나 Stage1의 학습에서 지나치게 오랜 시간이 걸려 Stage2(고화질의 이미지) 학습을 하지 못함

03 AttnGAN 모델 구현의 한계점

AttnGAN(Attention + StackGAN)을 활용하여 더 정밀한 결과를 얻을 수 있으나 시간 상의 문제로 StackGAN + WGAN-GP로 변경함

04 Mode collapse 해결의 한계점

- Mode collapse(모드 붕괴) 발생을 해결하기 위해 WGAN-GP 도입
- But, 모델이 지나치게 복잡해짐

THE

END

감사합니다
