

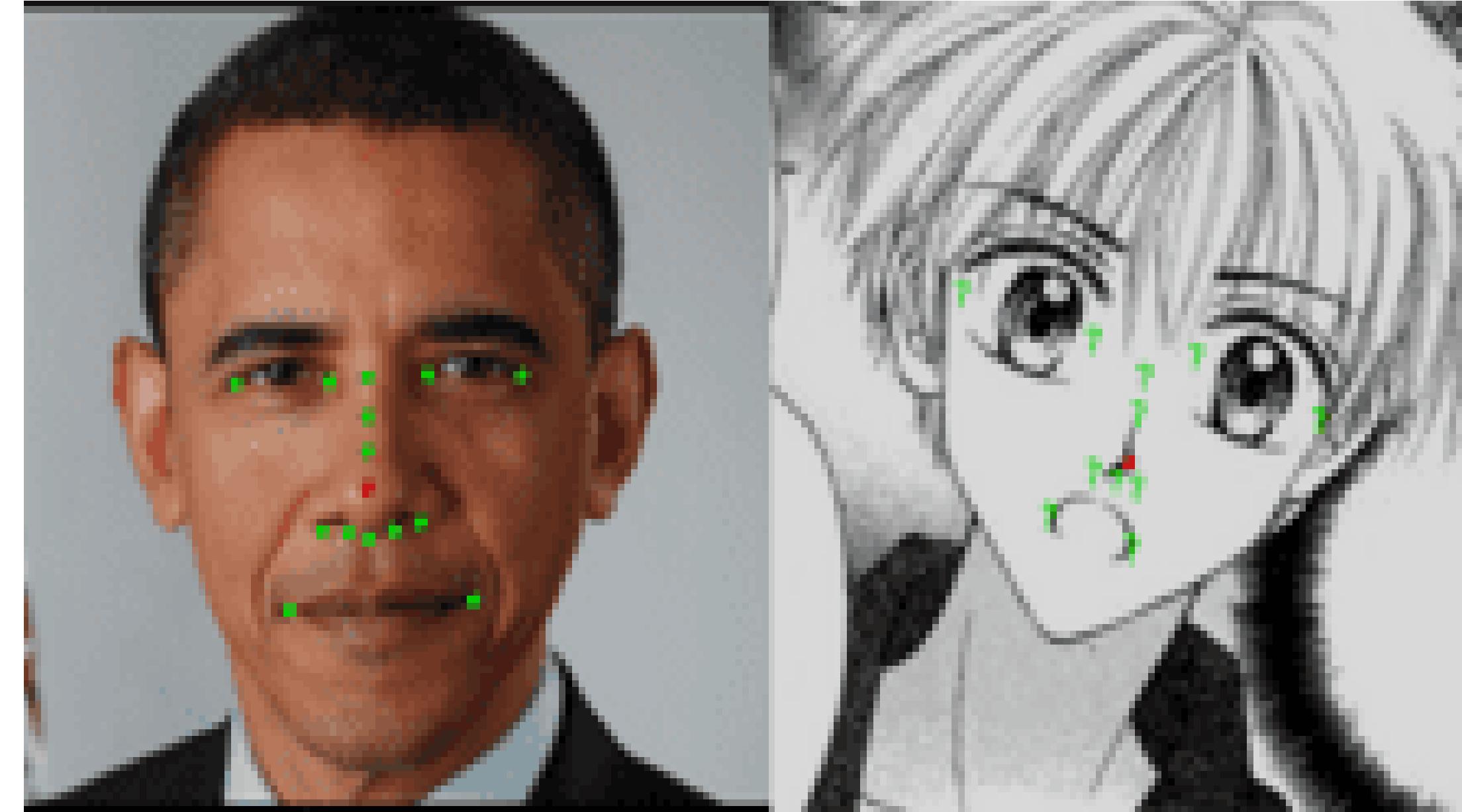
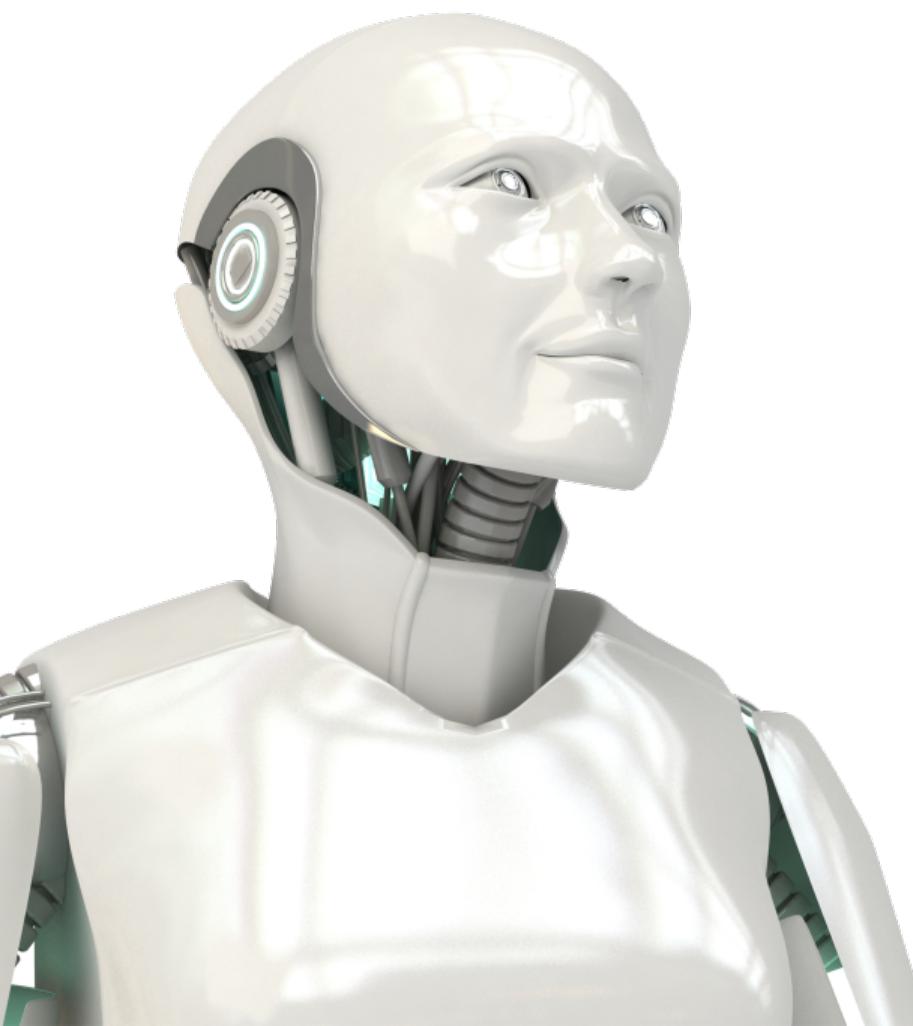
Character Facial Landmark Detection

시각인공지능
프로젝트 최종발표

조영진 (20162539)
나요셉 (20182800)
이동훈 (20173024)

CONTENTS

사람의 얼굴을 대상으로 **랜드마크** 탐지한 모델로
다양한 **캐릭터**에 적용시켜본다.



01 중간발표 요약

연구주제
공개코드 설명
데이터셋 설명

02 데이터 수집 및 변경

Manga 109 세부사항
Matching Program
XML 변환 및 생성

03 모델 성능 비교

다양한 모델 적용
전이학습 적용

04 Inference

애니메이션 캐릭터 적용
최종 모델 선정
결과

중간발표 요약

연구주제

사람의 눈, 코, 입, 눈썹, 턱선과 같은 사람의 얼굴을 대표하는 랜드마크를 추출해내는 방법

딥러닝 기반의 전략 등장으로 Facial Landmark Detection이 발전하기 시작

클래식 알고리즘 vs SOTA 성능비교 (Comparison in normalized mean error on the 300W)

Method	Common	Challenging	Fullset
Inter-pupil Normalization (IPN)			
RCPR [4]	6.18	17.26	8.35
CFAN [42]	5.50	16.78	7.69
ESR [5]	5.28	17.00	7.58
SDM [38]	5.57	15.40	7.50
LBF [24]	4.95	11.98	6.32
CFSS [46]	4.73	9.98	5.76
3DDFA [48]	6.15	10.59	7.01
TCDCN [45]	4.80	8.60	5.54
MDM [29]	4.83	10.14	5.88
SeqMT [12]	4.84	9.93	5.74
RAR [37]	4.12	8.35	4.94
DVLN [35]	3.94	7.62	4.66
CPM [33]	3.39	8.14	4.36
DCFE [30]	3.83	7.54	4.55
TSR [22]	4.36	7.56	4.99
LAB [34]	3.42	6.98	4.12
PFLD 0.25X	3.38	6.83	4.02
PFLD 1X	3.32	6.56	3.95
PFLD 1X+	3.17	6.33	3.76
Inter-ocular Normalization (ION)			
PIFA-CNN [15]	5.43	9.88	6.30
RDR [36]	5.03	8.95	5.80
PCD-CNN [19]	3.67	7.62	4.44
SAN [9]	3.34	6.60	3.98
PFLD 0.25X	3.03	5.15	3.45
PFLD 1X	3.01	5.08	3.40
PFLD 1X+	2.96	4.98	3.37

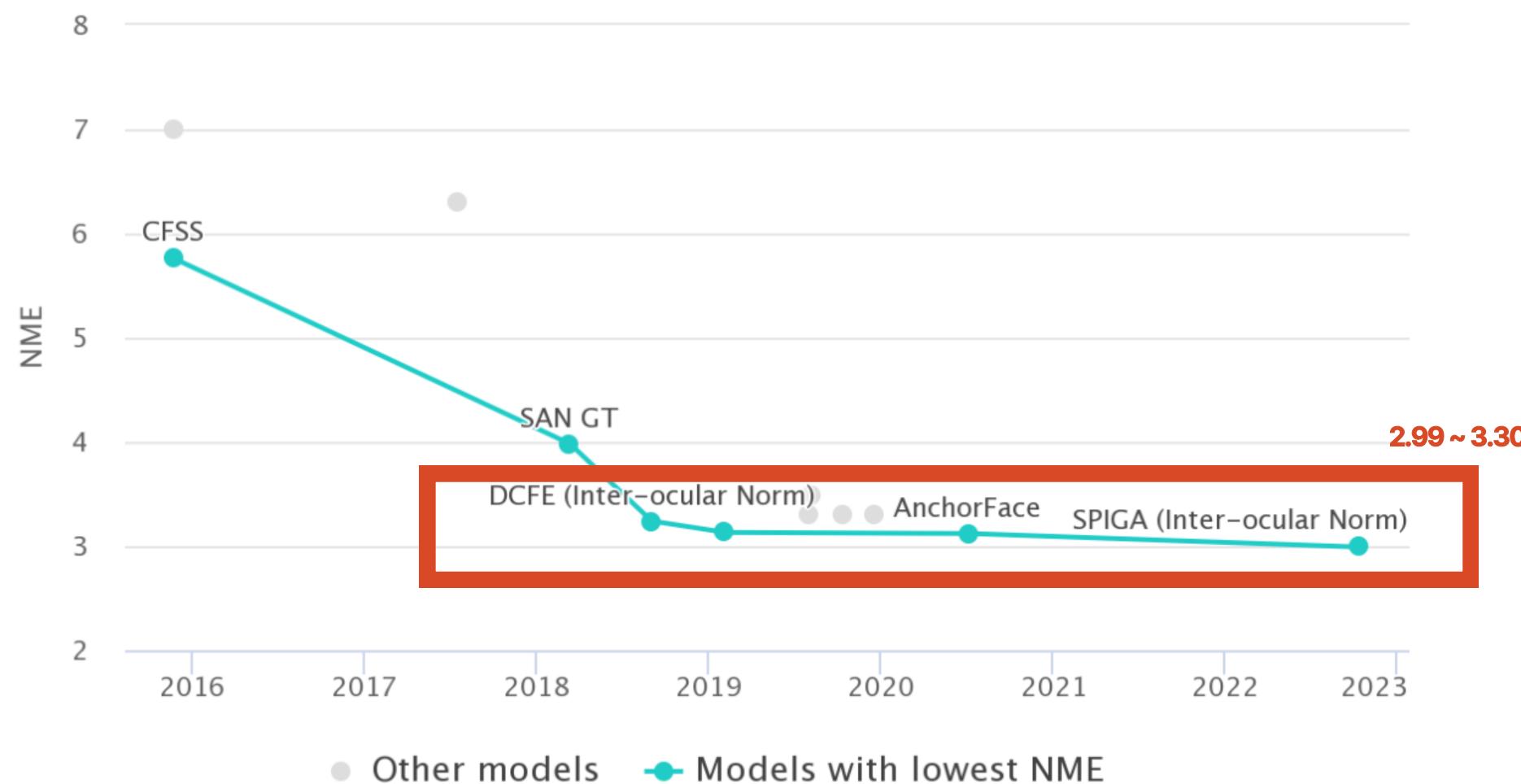
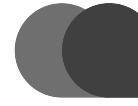


Table 4: Comparison in normalized mean error on the 300W Common Subset, Challenging Subset, and Fullset.



중간발표 요약

DataSet

iBUG 300-W dataset:

- 300개의 실내 및 300개의 실외 야생 이미지로 구성된 얼굴 dataset
- 68개의 랜드마크 포인트를 가지고 있음



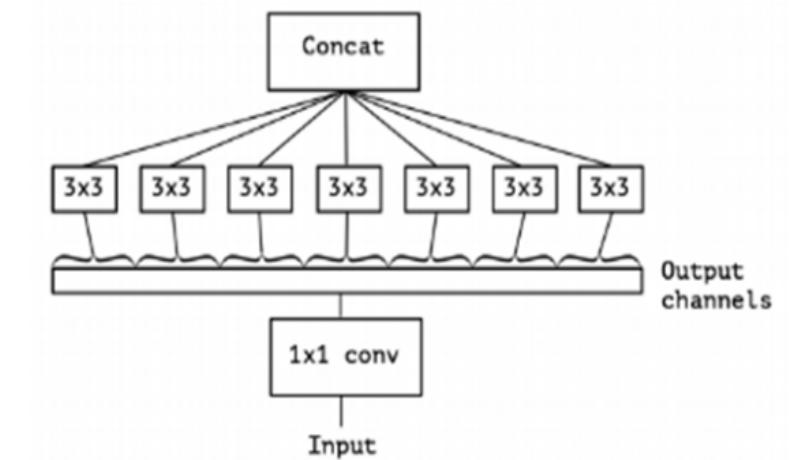
Augmentation

- 1) 랜덤 색상(hue), 채도(saturation), 명도(brightness), 대조(contrast)
- 2) Random Rotation
- 3) 이미지를 bounding box 좌표로 crop 시킴.



Xception

- 1) Xception은 공간정보를 더 남김없이 학습할 수 있습니다.
- 2) Xception은 Depthwise Seperable Convolution으로 더욱 효율적으로 연산할 수 있습니다.



<Xception 모듈>

중간발표 요약

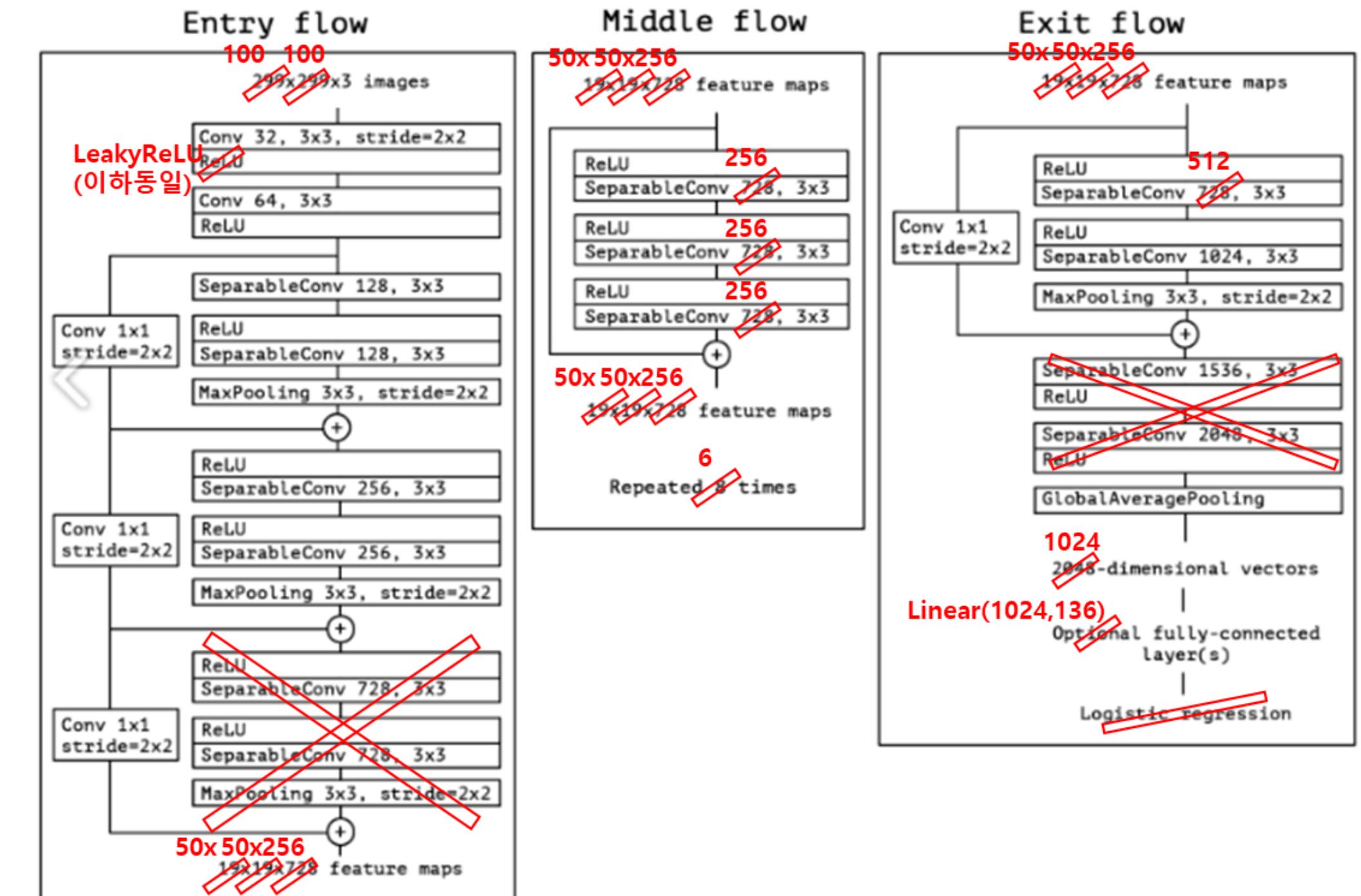
공개코드 설명

1. 논문에서 제시한 Xception 모델을 경량화 했습니다.

-> 일부 층을 없애고 Middle block의 수를 8개에서 6개로 줄였습니다.

2. 예측해야 하는 Landmark 68개 * 2 = 136개의 좌표를 예측하는 회귀 문제로 문제가 재정의 됩니다.

Figure 5. The Xception architecture: the data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Note that all Convolution and SeparableConvolution layers are followed by batch normalization [7] (not included in the diagram). All SeparableConvolution layers use a depth multiplier of 1 (no depth expansion).



● 중간발표 요약

● 데이터셋 설명

1) Caricature Character

- 총 7,800 장의 2D 캐리커처 이미지와 68개의 랜드마크로 구성된 데이터셋
- 2D 캐리커처를 가지고 3D 캐리커처를 만드는 과정에서 랜드마크 탐지한 데이터셋



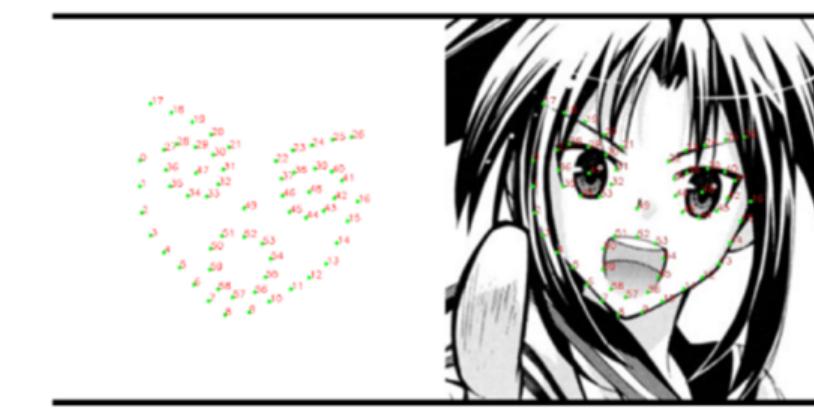
2) The Face of Art: Landmark Detection and Geometric Style in Portraits

- 총 그림 160장 자화상 이미지와 68개의 랜드마크로 구성된 데이터셋
- WikiArt 홈페이지에 있는 예술작품 중 16명의 각기 다른 화가들의 예술적 자화상



3) Manga109 Dataset

- 도쿄대학교 Aizawa Yamasaki Matsui 연구소에서 만든 만화 데이터셋
- 109권의 만화에서 만화책 정보와 만화책에 있는 얼굴 이미지를 추출한 데이터셋
- 이를 인용한 논문에서 수동으로 60개의 랜드마크 레이블 작업을 한 데이터셋



핵심 : 2D 캐리커처, 초상화, 만화 캐릭터 등 다양한 캐릭터 이미지를 학습하여 새로운 캐릭터들에도 효과적으로 적용할 수 있게 함

● 데이터 수집 및 변경

● Manga 109 Dataset

● Manga 109 Dataset의 필요

실제 애니메이션의 그림이 보통 만화가 원작이기 때문에
애니메이션 이미지에서 키포인트를 추출할 때 만화 데이터라고 판단

● Manga 109 Dataset의 문제

Image의 annotation과 Landmark Image 가 매칭되지 않음

<Landmark Data File 명 예시>

user0_MiraiSan_009_face01_x131y48w121h130.json

- 1) MiraiSan : 책 제목
- 2) 009 : 페이지
- 3) face 01 : Image annotation 의 face 번호
- 4) bounding box 좌표

manga



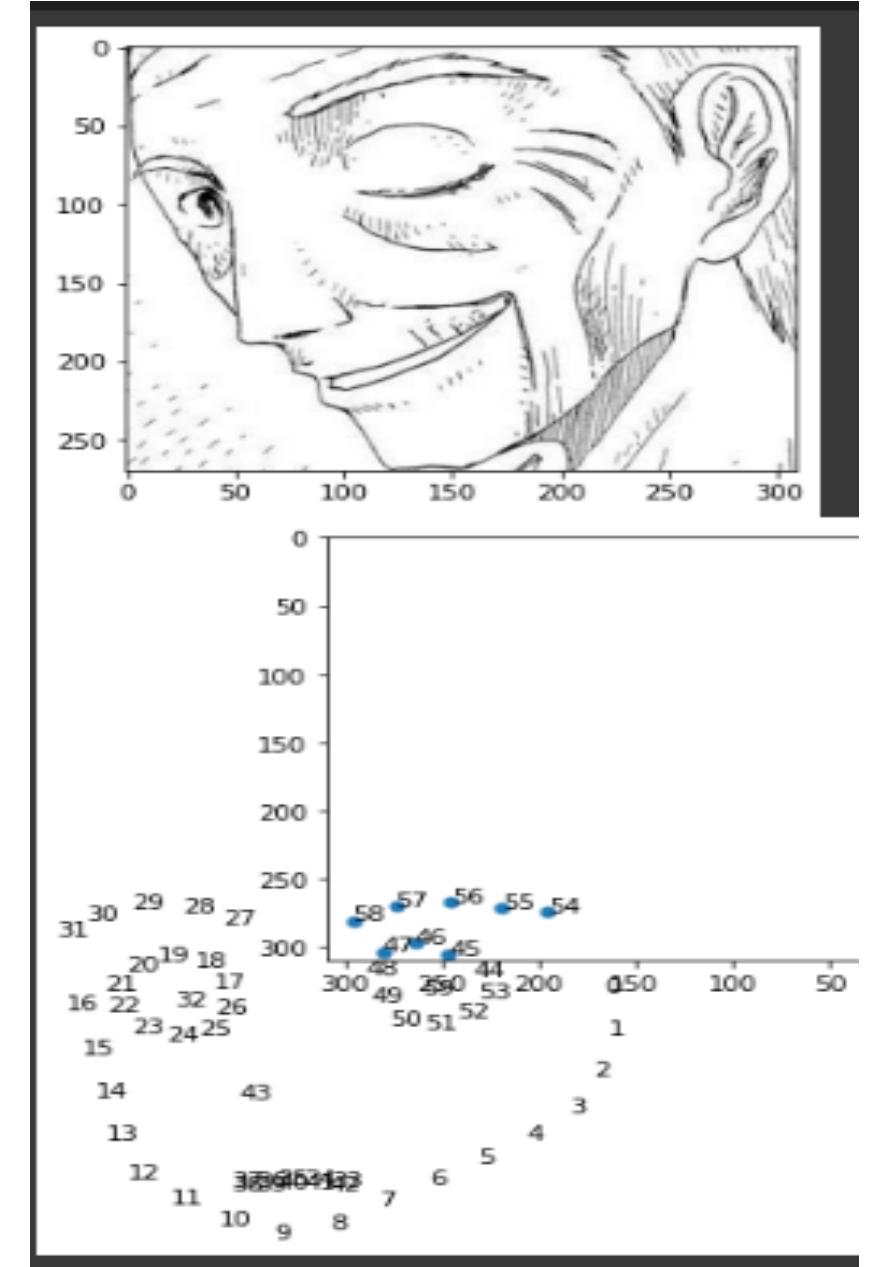
anime



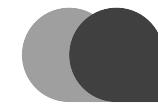
데이터셋 수집 및 변경

```
<?xml version="1.0"?>
- <book title="MiraiSan">
  + <characters>
    - <pages>
      <page height="1170" width="1654" index="0"/>
      <page height="1170" width="1654" index="1"/>
      + <page height="1170" width="1654" index="2">
      + <page height="1170" width="1654" index="3">
      + <page height="1170" width="1654" index="4">
      + <page height="1170" width="1654" index="5">
      + <page height="1170" width="1654" index="6">
      + <page height="1170" width="1654" index="7">
      + <page height="1170" width="1654" index="8">
    - <page height="1170" width="1654" index="9">
      <text id="0004019e" ymax="551" xmax="1090" ymin="169" xmin="928">次に目覚めたときはおまえの墓参りにも行</text>
      <body id="0004019f" ymax="734" xmax="1576" ymin="146" xmin="980" character="0004008c"/>
      <text id="000401a0" ymax="925" xmax="1069" ymin="791" xmin="1019">また8世代ほど寝るそうです</text>
      <frame id="000401a1" ymax="1088" xmax="1324" ymin="751" xmin="907"/>
      <text id="000401a2" ymax="260" xmax="1535" ymin="143" xmin="1508">いやあた</text>
      <face id="000401a3" ymax="952" xmax="1179" ymin="918" xmin="1133" character="00040040"/>
      <frame id="000401a4" ymax="735" xmax="1576" ymin="84" xmin="905"/>
      <body id="000401a5" ymax="1085" xmax="1222" ymin="898" xmin="1119" character="00040040"/>
      <frame id="000401a6" ymax="1086" xmax="1577" ymin="751" xmin="1339"/>
      <face id="000401a7" ymax="565" xmax="1424" ymin="295" xmin="1115" character="0004008c"/>
```

Landmark Data File 예시와 매칭되는 Image Annotation



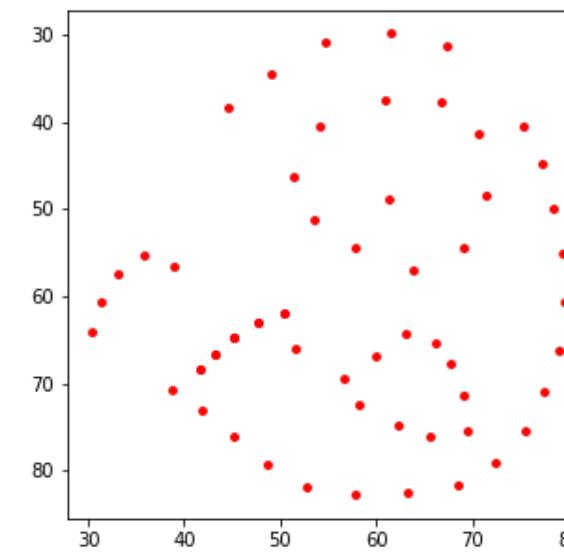
매칭 결과



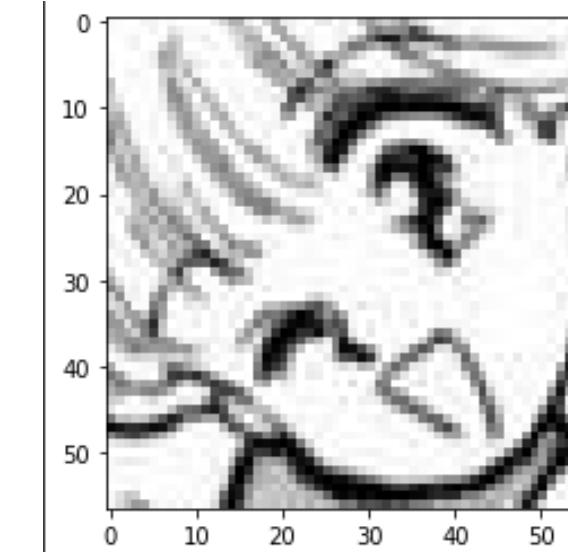
데이터셋 수집 및 변경



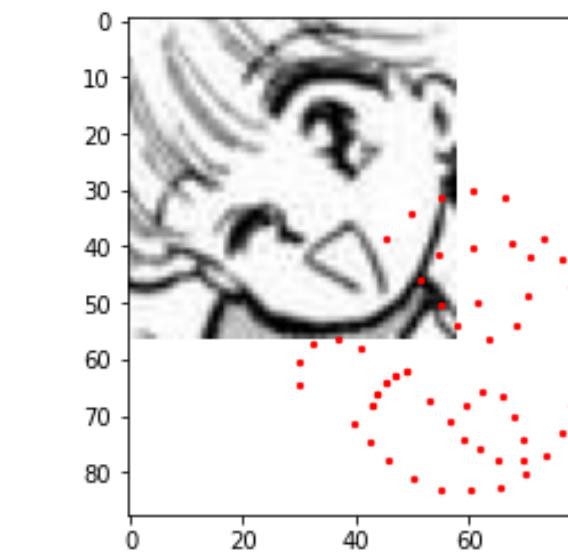
Matching program



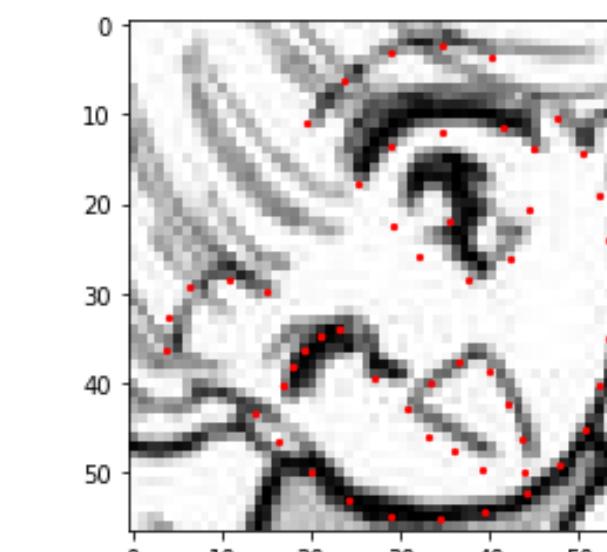
<랜드마크 예시 >



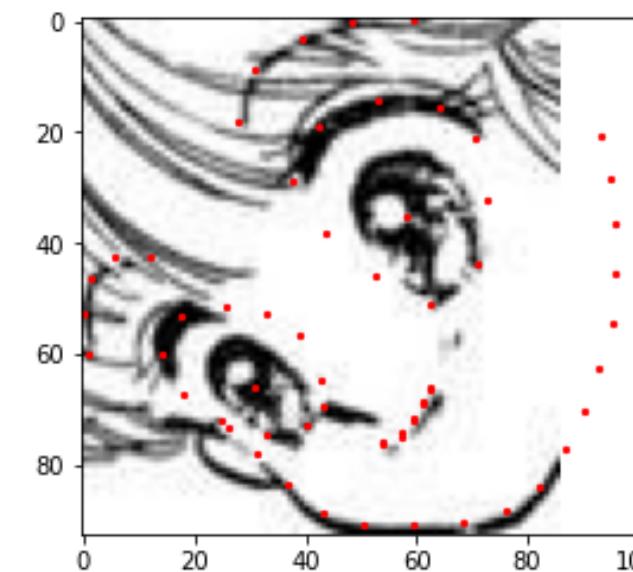
<Manga 109 cropped 얼굴 image 예시 >



<동시 plot >



<랜드마크 위치 조정 이미지 >



<다른 이미지 조정 >

데이터셋 수집 및 변경

Matching program

1. 전체 이미지에서 후보 이미지 선별

crop된 얼굴 이미지가 1500장 이상이기 때문에 일일이 대조하기엔 어려움이 있음.

2. 후보 이미지 선별 과정

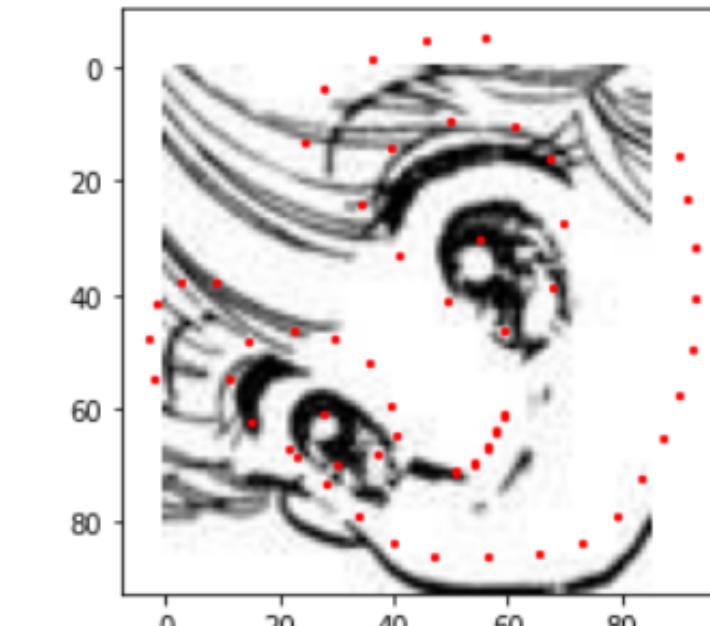
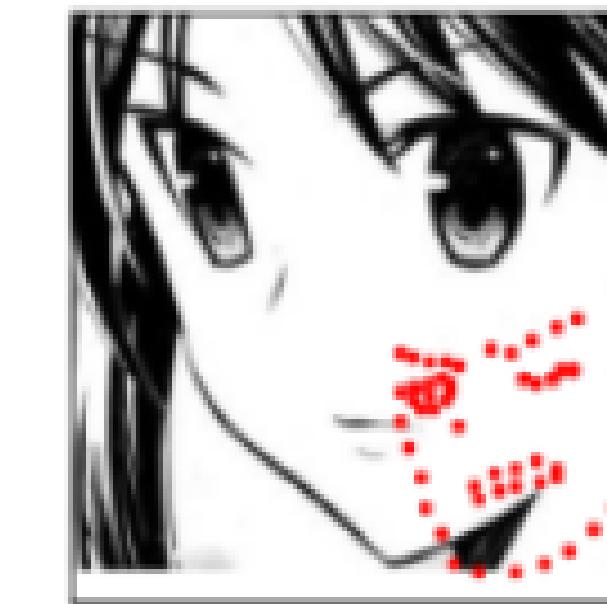
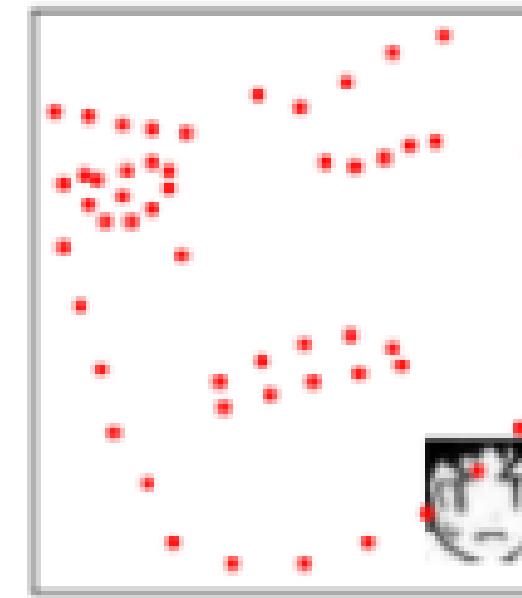
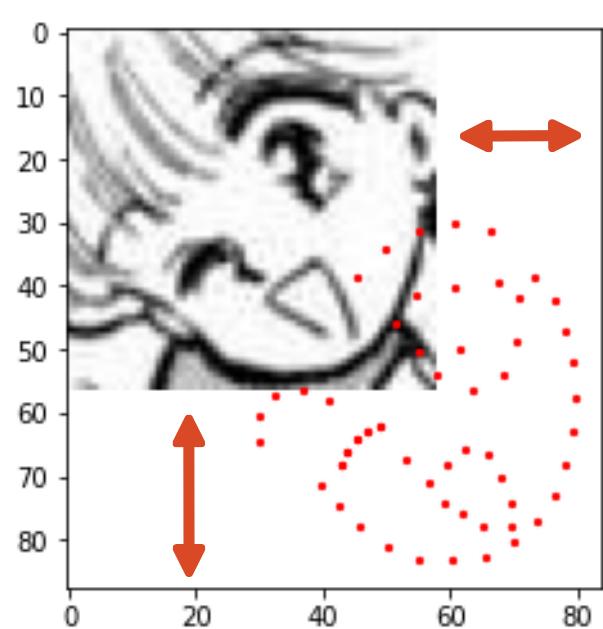
2-1) 랜드마크의 위치 max값에서 이미지 shape을 빼줌 -> 이미지와 랜드마크 사이의 여백 제거

2-2) 조정한 랜드마크 위치가 0 보다 작은 경우 제거

2-3) min이 전체 이미지 1/3 지점보다 큰 이미지 제거

3. 이미지와 랜드마크 매칭

선별된 이미지 2,300장 각각 plot, 수작업으로 고르고 x,y 좌표 수동 조정해서 이미지와 랜드마크 위치를 정확하게 매칭



2-1

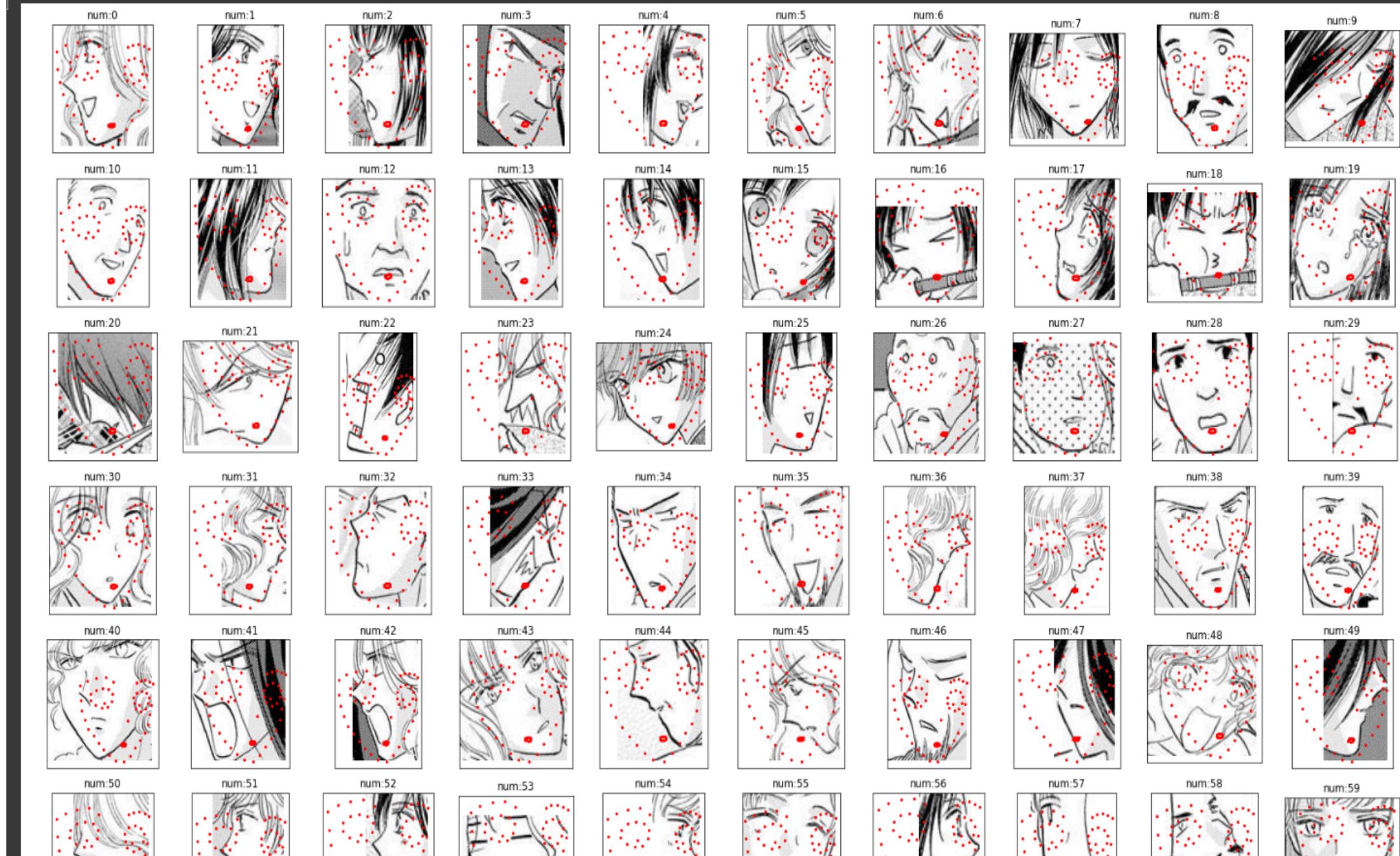
2-2

2-3

3

데이터셋 수집 및 변경

Matching program



후보 이미지들과 landmark 매칭



x, y 좌표 이동 후 저장

● 데이터셋 수집 및 변경

● XML 변환 및 생성

기존 코드와 같이 새로운 데이터셋을 xml로 생성하여 입력 형식을 같게함

▷ 어떤 데이터셋이든 xml만 만들 수 있으면 그 외 모든 코드는 수정할 필요 없음

[1번째 데이터셋] 현재 데이터셋 형식 image + .npy

```
# 변경 형식 xml (image_path + landmark(x,y))
```



[2번째 데이터셋] 현재 데이터셋 형식 image + .json

```
# 변경 형식 xml (image_path + landmark(x,y))
```



[3번째 데이터셋] 현재 데이터셋 형식 .npy+ .npy

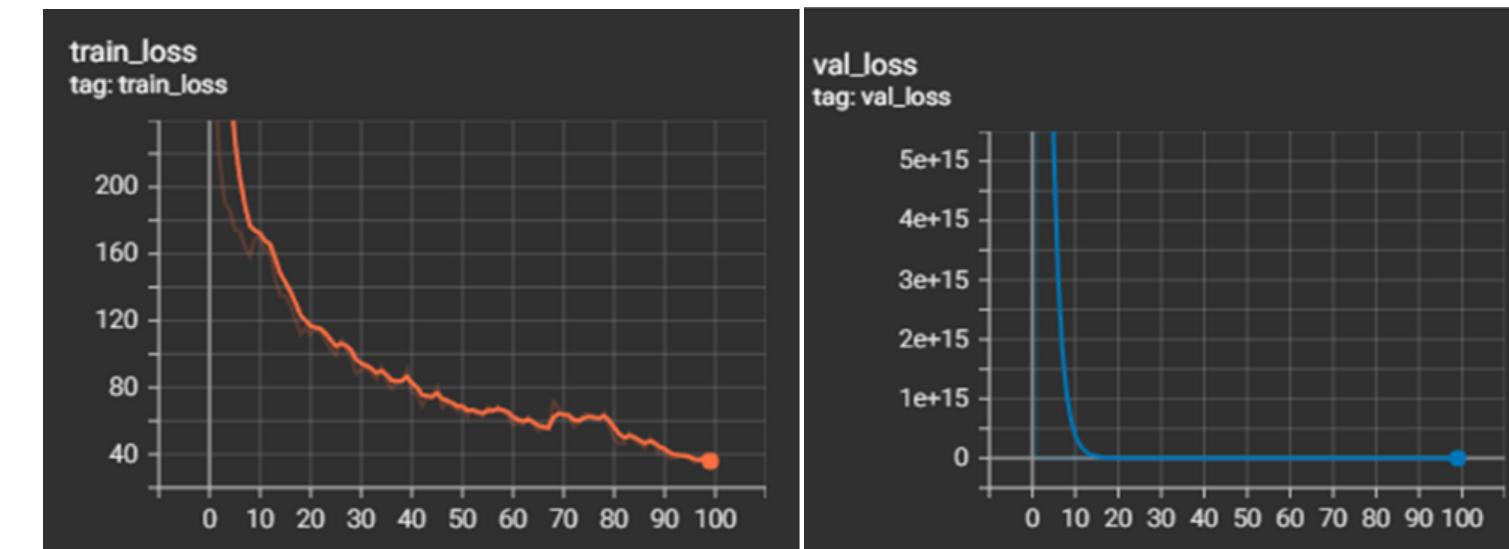
```
# 변경 형식 xml (image_path + landmark(x,y))
```



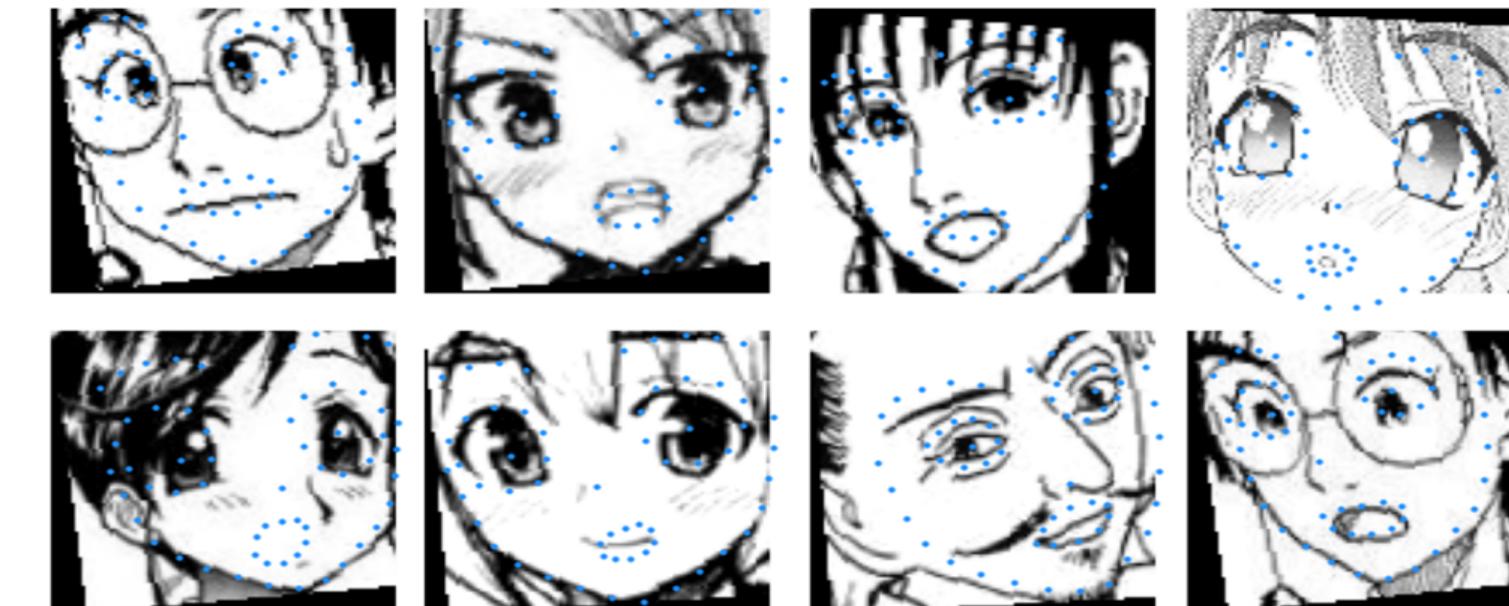
모델 성능 비교

다양한 모델 적용

model	train loss	val loss
Xception	42	229
Inception	206	117
Resnet 50	32	29
Resnet 50 (Adaloss)	10	13
VGG 16	235	208



Resnet 50 loss graph

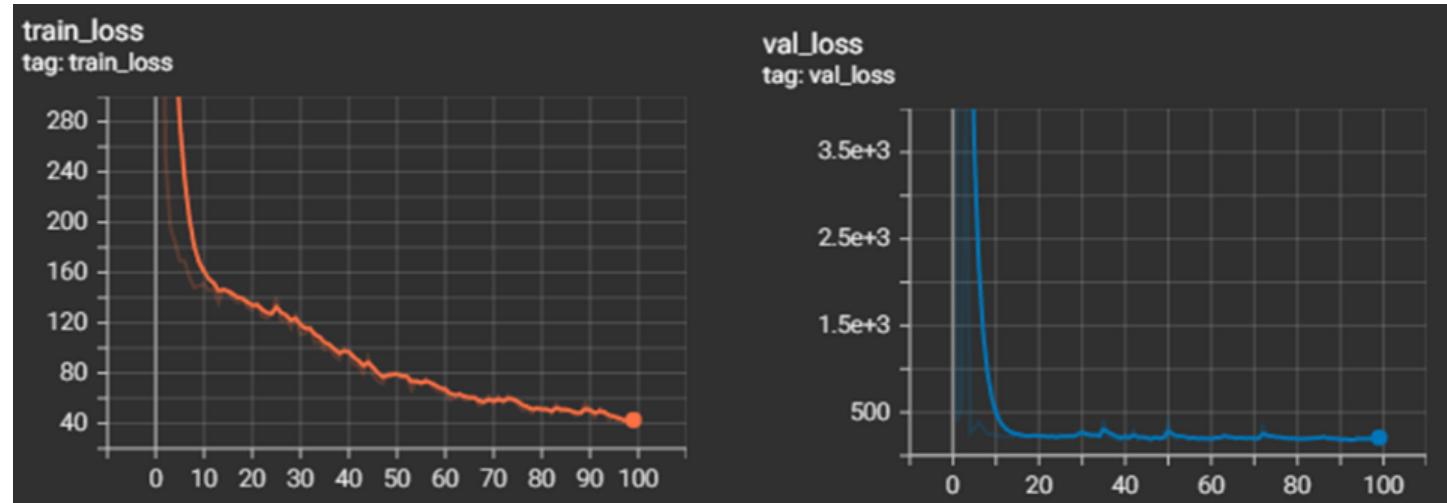


결론 : 전반적인 얼굴 랜드마크들을 잘 탐지한다.

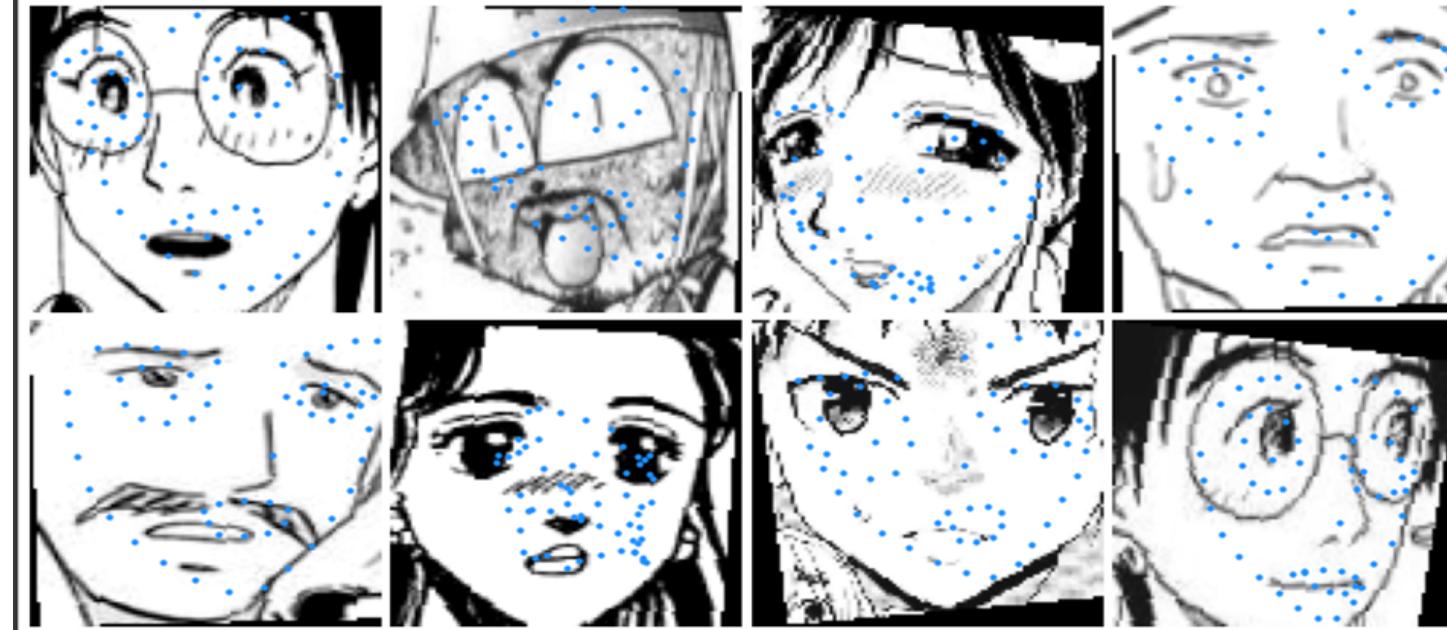
- ▶ loss를 비교했을 때 다섯 가지 모델 중에 Resnet 50이 가장 좋은 성능을 보인다.

● 모델 성능 비교

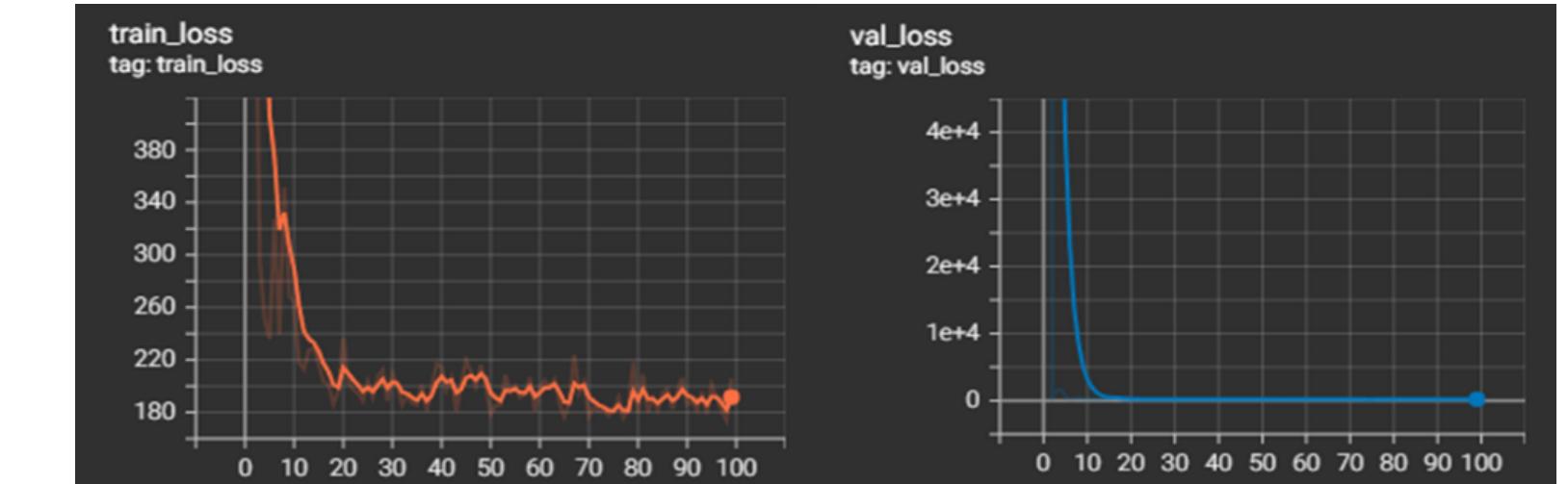
● 다양한 모델 적용 - 성능 그래프 및 validation



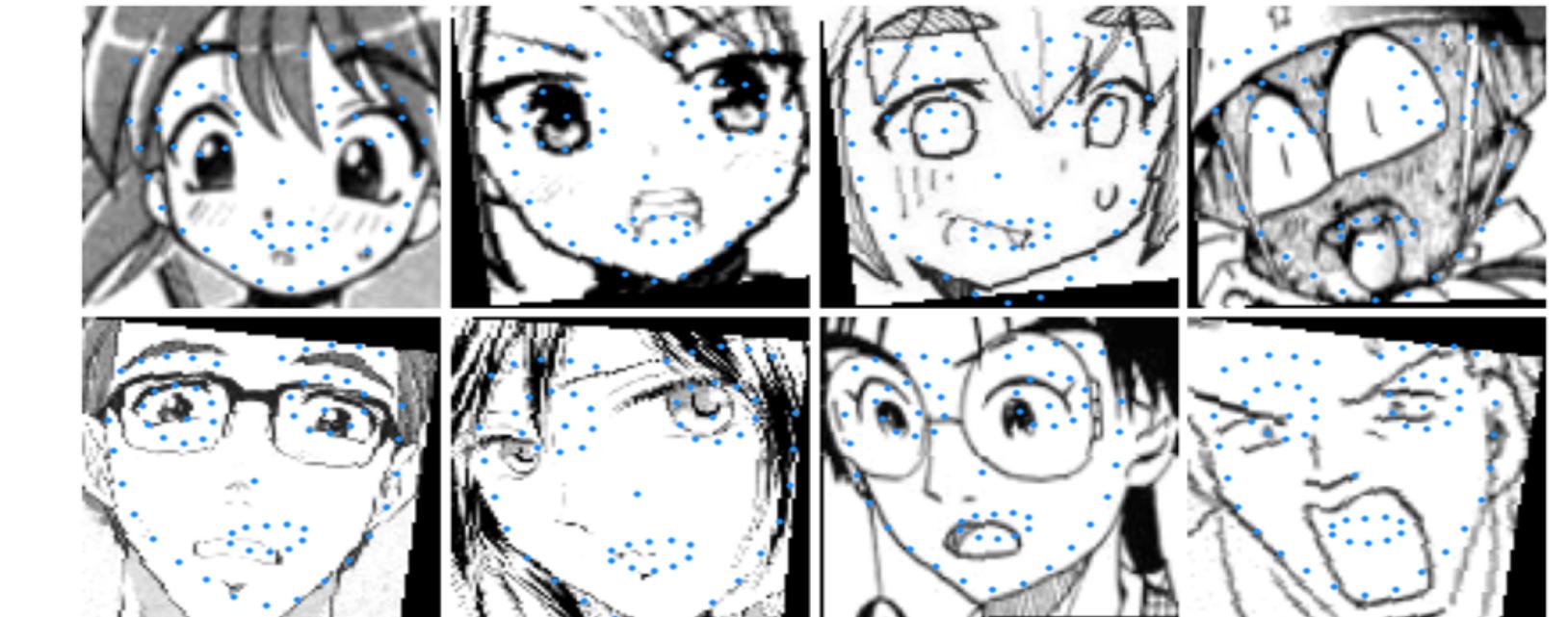
Xception loss graph



결론: 얼굴의 전체 형태는 잘 잡지만, 이미지 shifting이 심하고 scale도 안맞다.



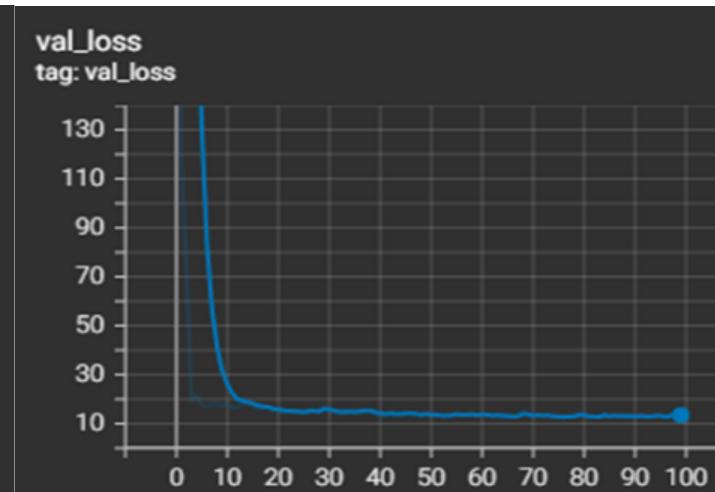
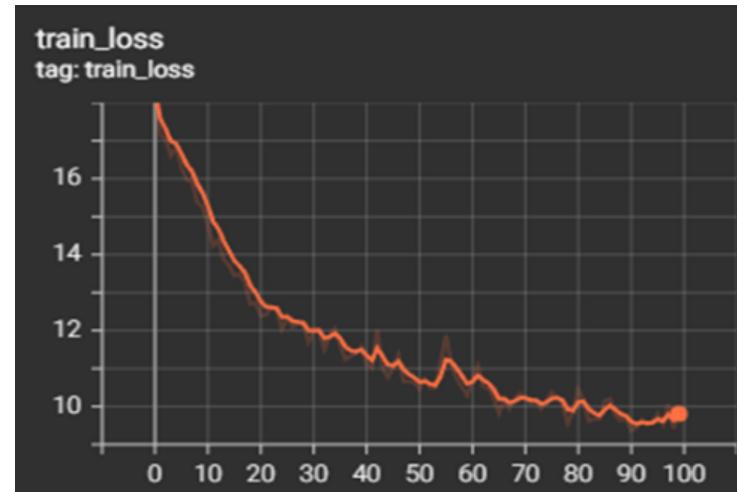
Inception loss graph



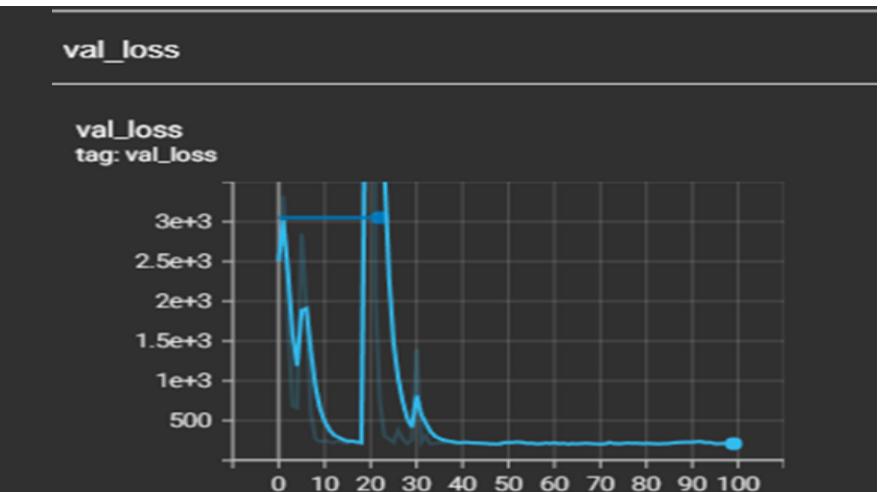
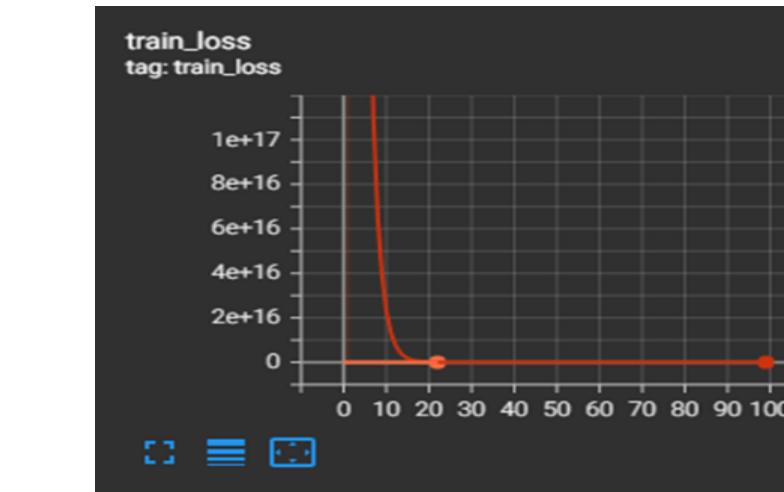
결론: 이미지 각각의 특징을 잘 캐치하지 못하고 비슷한 모형으로 찍힌다.

● 모델 성능 비교

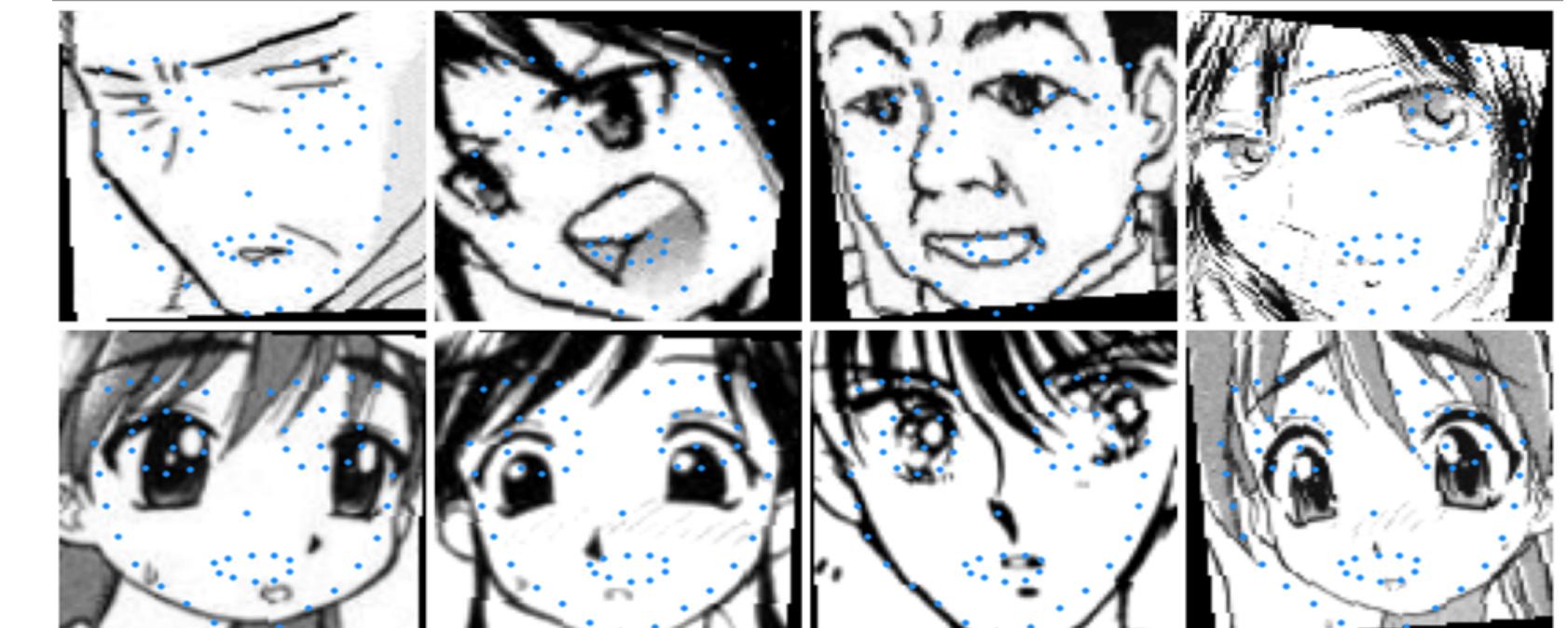
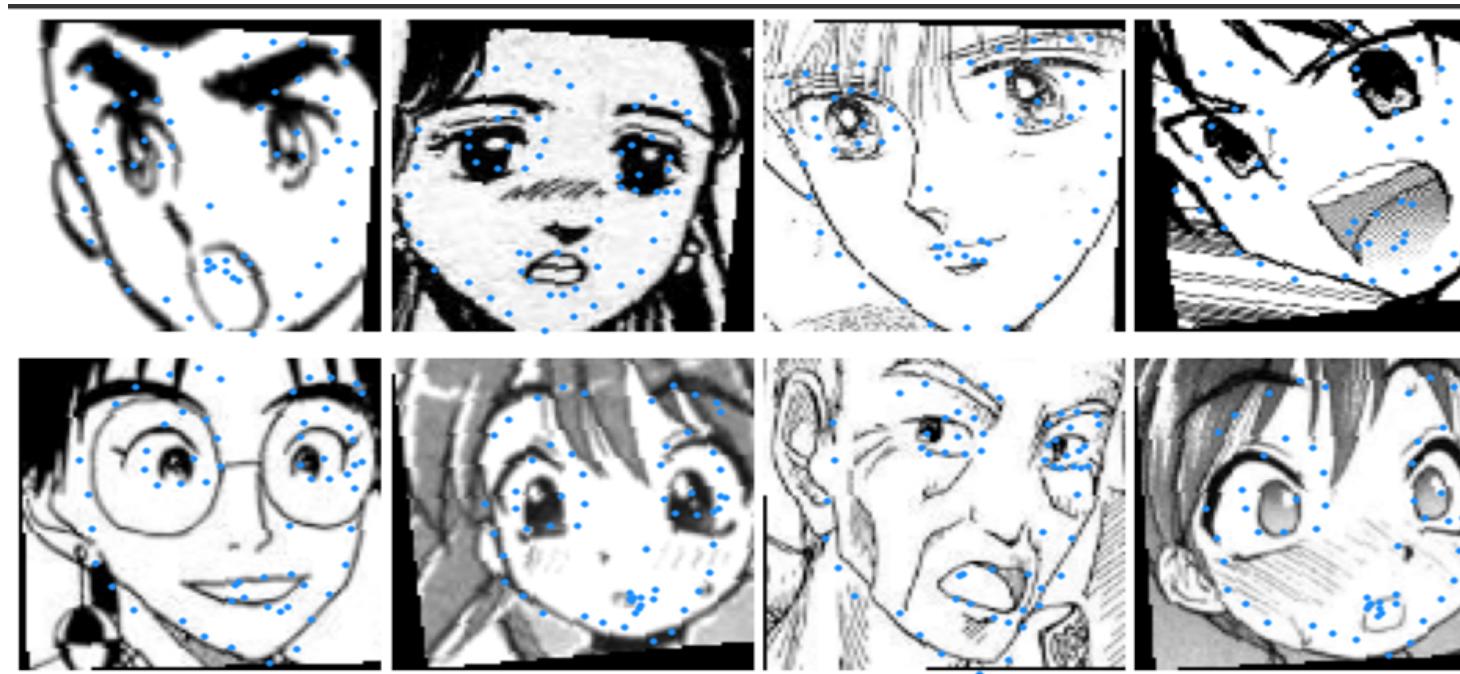
● 다양한 모델 적용 - 성능 그래프 및 validation



Resnet 50 with Adaloss graph



VGG 16 loss graph



결론: 얼굴 형태를 유지하지 못한 채로 마구잡이로 찍는다.

결론: 이미지 각각의 특징을 잘 캐치하지 못하고 비슷한 모형으로 찍힌다.

● 모델 성능 비교

● 전이학습 적용

Manga Image가 너무 적기 때문에 다른 Dataset에서 잘 적용되는 모델을 활용하여 성능을 높이는게 목표

전략1. 모든 Layers Freeze

본래 dataset 300W를 학습한 Xception 가중치를 freeze 시켜, Portraits dataset에 적용
-> 아래 그림 처럼 전혀 landmark를 검출하지 못함



● 모델 성능 비교

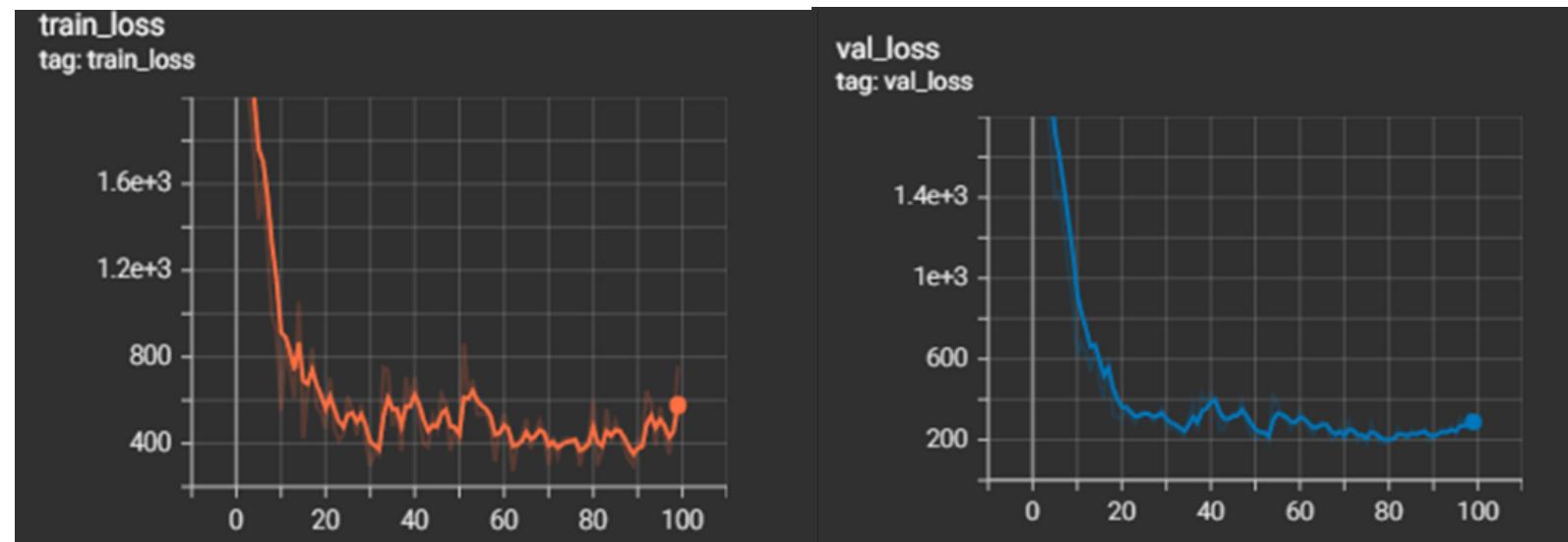
● 전이학습 적용

전략2. Feature Extractor Freeze, Classifier Train

본래 데이터 셋 300W로 학습한 모델의 Feature Extractor만 Freeze,

Classifier 136 -> 120 변경 후 학습

-> 얼굴 형태만 간신히 유지



● 모델 성능 비교

● 전이학습 적용

전략3. 모든 Layers Train

학습 데이터가 적기 때문에 모든 Layer를 Train시키는 방법이 성능이 높다고 판단

-> Entry Block Weight로 Initialization하는 방법이 가장 좋은 성능 기록

Initialization	train loss	val loss
Entry	51 (50)	43 (28)
Entry + Middle	41	116
Entry + Middle + Exit	37	92



Entry



Entry + Middle



Entry + Middle + Exit

Inference

● 애니메이션 캐릭터에 적용

약 3초의 GIF를 약 15FPS의 Image 파일로 저장한 뒤 이미지 파일 각각에 대한 Face Landmark Detection 진행 후 다시 GIF로 변환하였다

● 최종 모델 선정

학습 과정

- 1) Xception : Caricature Data 학습 weight의 Entry Block 만 초기화하여 Manga Data 학습
- 2) ResNet50 : Caricature Data 학습 weight 전부를 초기화하여 Manga Data 학습
- 3) Inception : Caricature Data 학습 weight 전부를 초기화하여 Manga Data 학습

모델	Train Loss	Val Loss
Xception	56	26
ResNet50	33	21
Inception	54	18



Xception -> Resnet -> Inception 순으로 gif에 좋은 결과가 있기를 기대



1) Xception : <https://youtube.com/shorts/Z8F9Qo7AgeE?feature=share>

2) ResNet50 : <https://youtube.com/shorts/2m9TmERGR78>

3) Inception : <https://youtube.com/shorts/ko7mzJfTxfw>

감사합니다