

# 나만의 여자 아이돌 만들기

# AutoEncoder # KNN

# Pix2PixHD # CycleGAN # U-GAT-IT

CV-12  
(CV코인 팀)

**boostcamp** ai tech



## 캠퍼 소개



T\_2097 박진한

# Stage 1 # KNN # 알고리즘



T\_2141 유형진

# Pix2PixHD # stage 1



T\_2162 이양재

# Data # Preprocessing



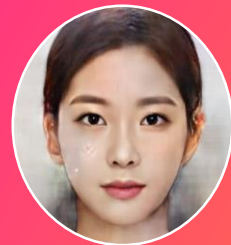
T\_2183 임채영

# Web serving # Front-end



T\_2232 최태종

# Web serving # Back-end  
# Pix2PixHD



T\_2236 한재현

# CycleGAN # U-GAT-IT



# CONTENTS

- 1 INTRODUCTION
- 2 FACING CHALLENGES
- 3 DATA
- 4 MODEL
- 5 SERVING
- 6 PROBLEM & SOLUTION
- 7 FURTHER WORK

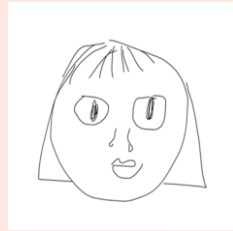
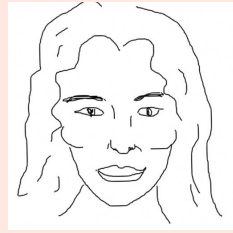
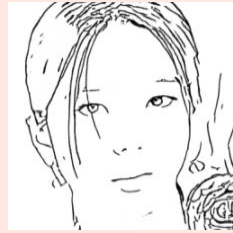
# 1 Introduction

---

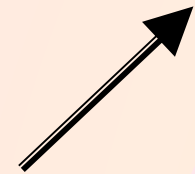
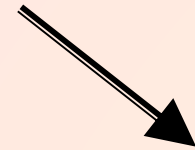
- ▲ 주 제: 스케치로 나만의 여자 아이돌 만들기
- ▲ 선정 배경: GAN 구현 및 프로젝트의 흥미성
- ▲ 목 표: 사용자가 그려 넣은 스케치의 특징을 포함한 여자 아이돌을 생성하는 모델을 서비스를 제공

# 2 Facing challenges

## 1) Data



## 2) What if?



# 3.1 Data



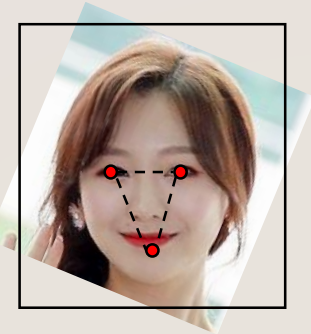
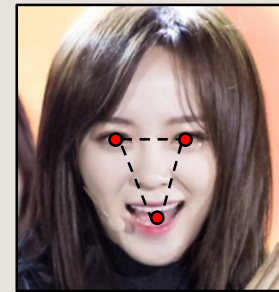
Collecting

- ◆ AIHUB  
: 1669 images
- ◆ Crawling idol images  
: 2449 images

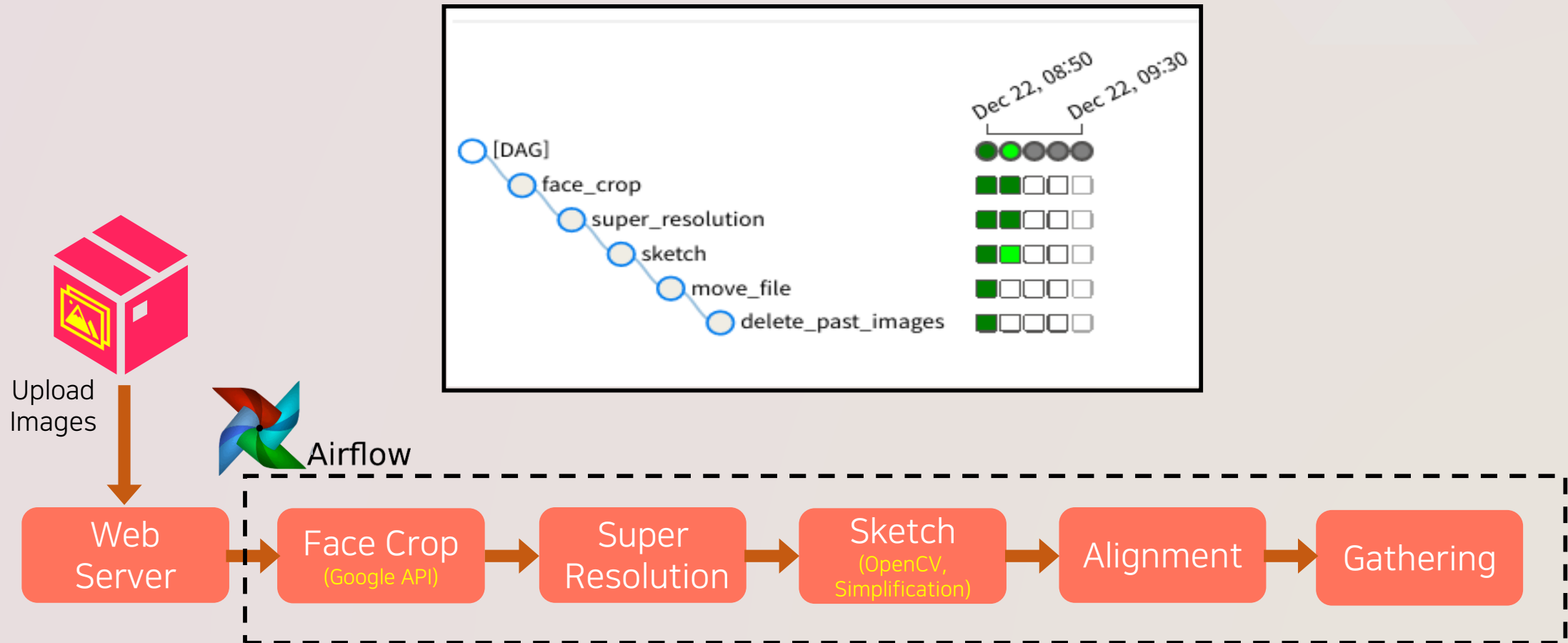


Filtering

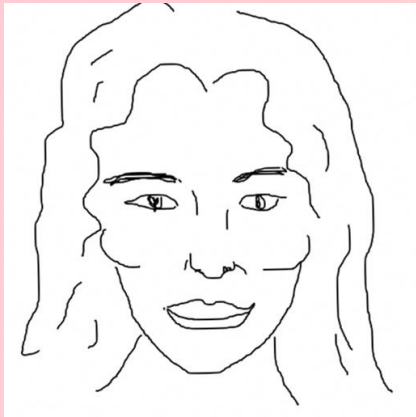
- ◆ Frontal faces
- ◆ General images



# 3.2 Data Pipeline

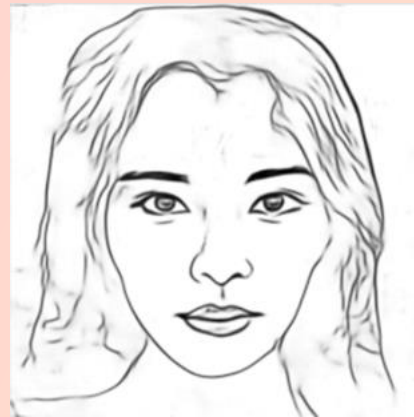


# 4.1 Whole Pipeline



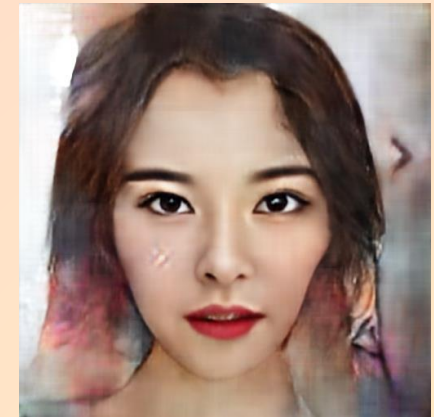
input sketch

Stage 1



refined sketch

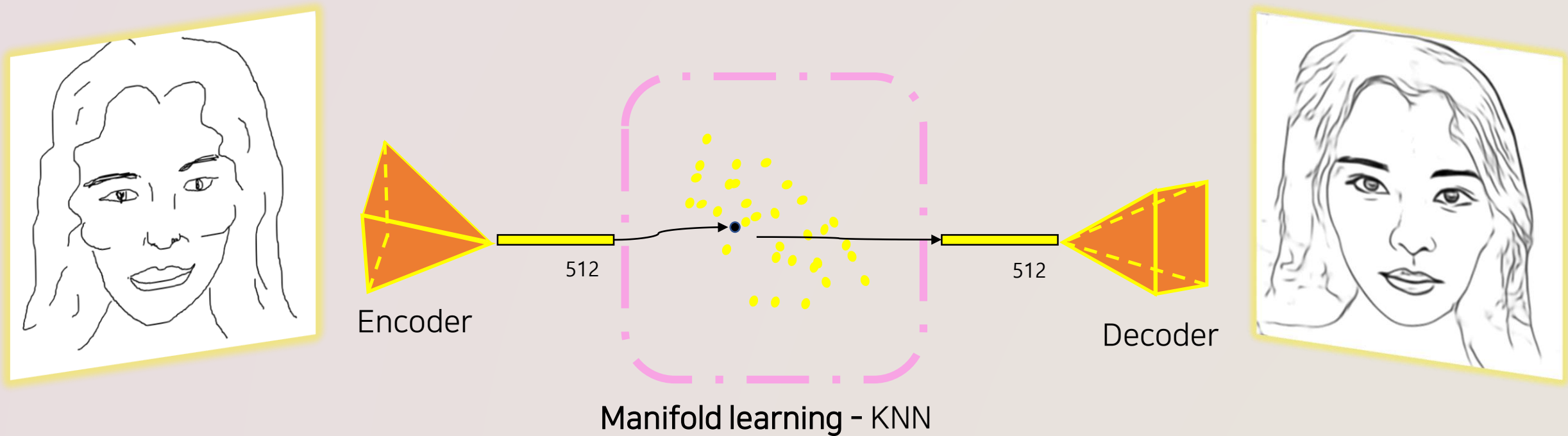
Stage 2



output image



# 4.2 Stage 1 - Autoencoder



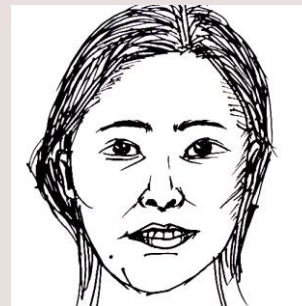
# 4.2 Stage 1 – k of KNN

## ▲ feature vector space (512 dim)

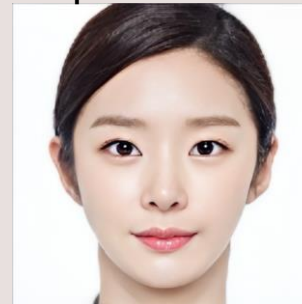
1. Select k-nearest neighbors

2. Find weights

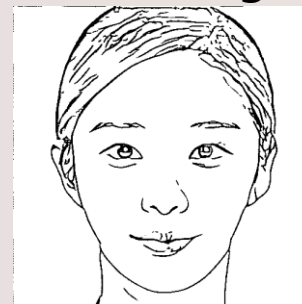
3. Get weighted average feature vector



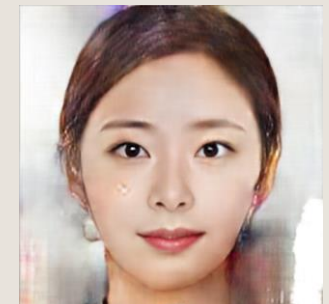
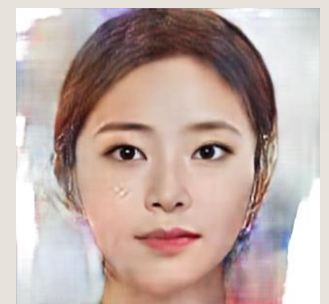
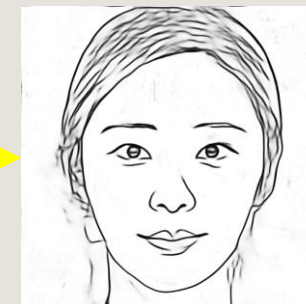
Input Sketch



Train image



Train Sketch

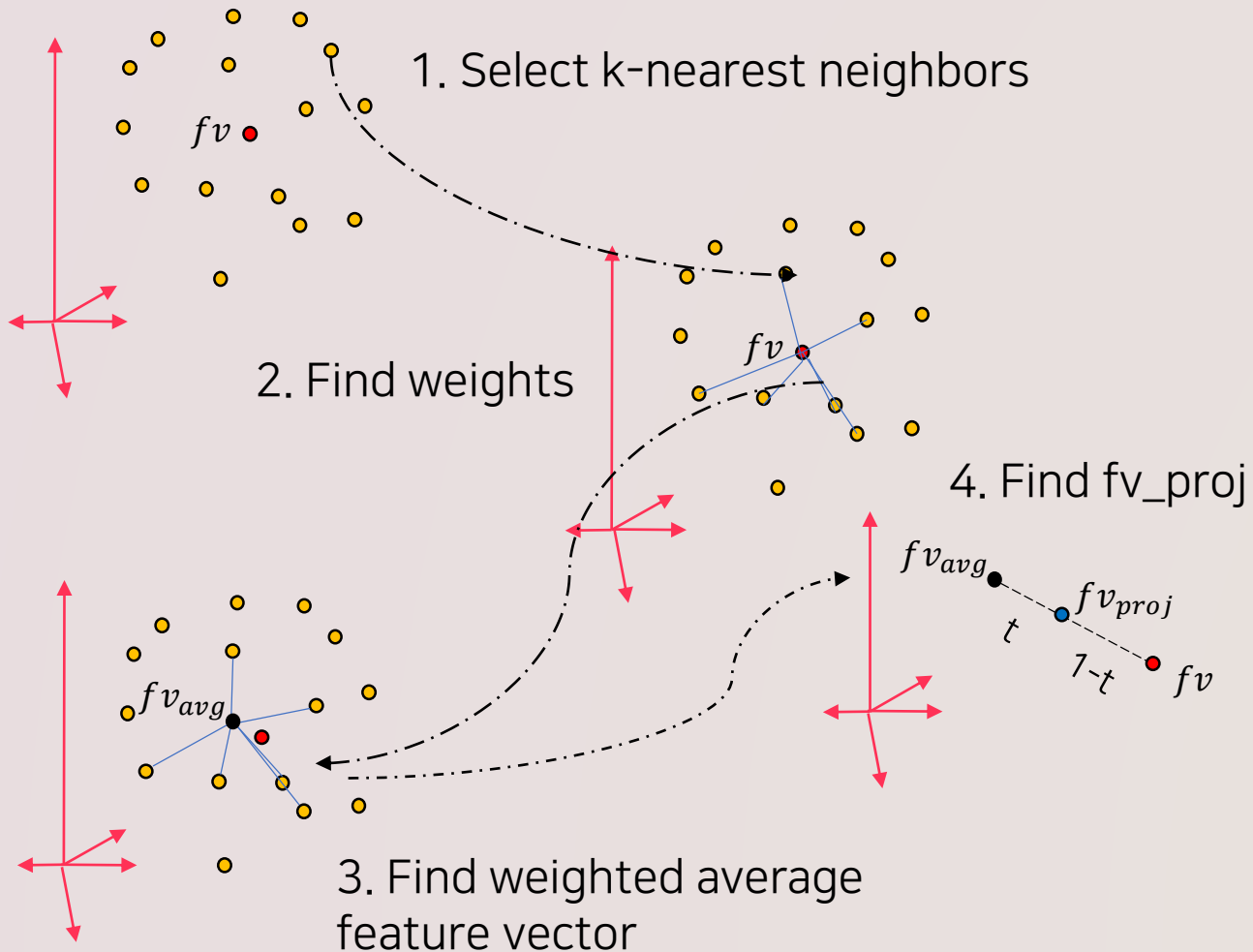


K  
10  
5  
3

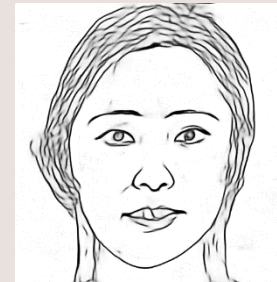
Too similar

# 4.2 Stage 1 – t of KNN

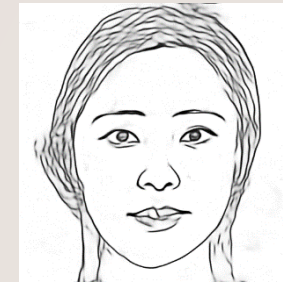
## ▲ feature vector space (512 dim)



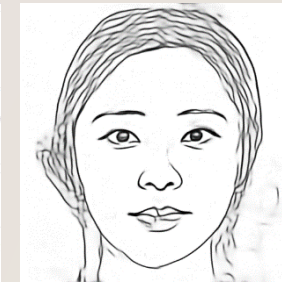
Input Sketch



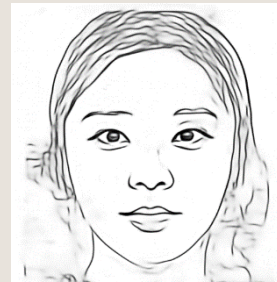
$t=1.0$



$t=0.7$

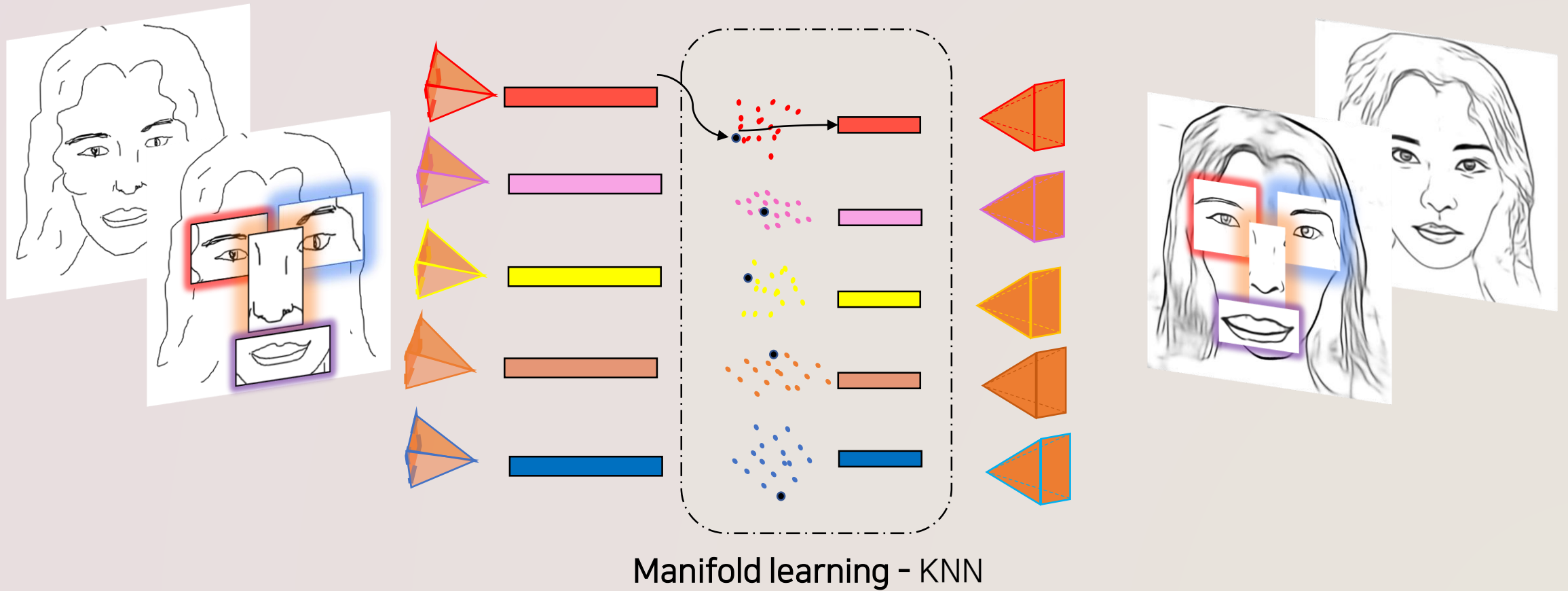


$t=0.4$



$t=0.0$

# 4.2 Stage 1 – number of Autoencoder



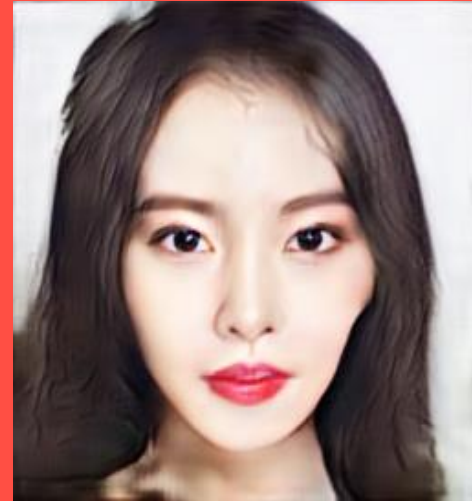
# 4.3 Stage 2 – Image to Image Translation



UnPaired



CycleGAN



U-GAT-IT

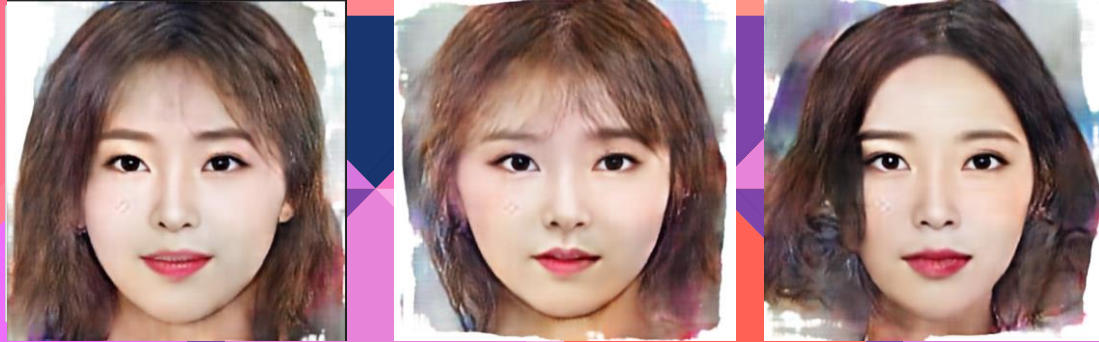
Paired



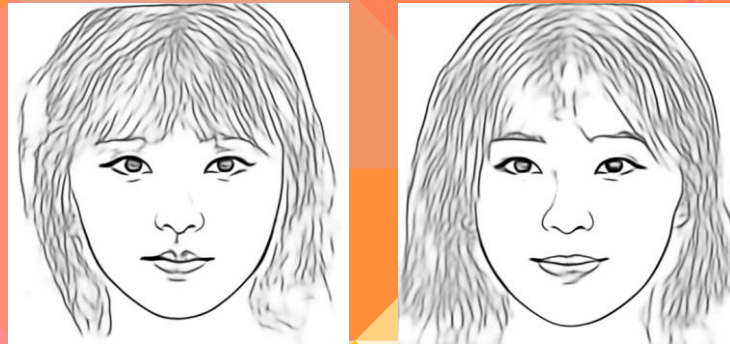
Pix2PixHD



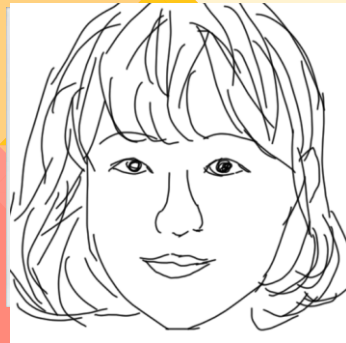
# 4.4 Result



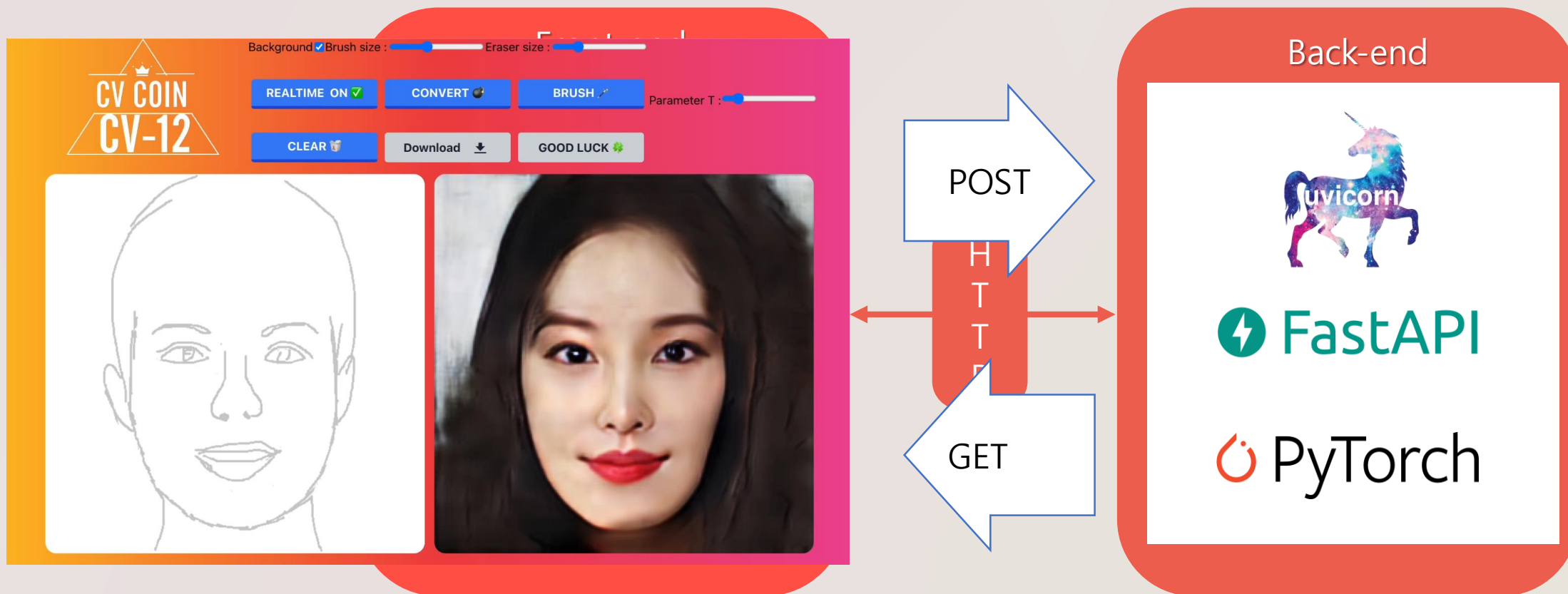
**STAGE 2**



**STAGE 1**



# 5 Serving





Background ☒

Brush size :



Eraser size :



REALTIME ON ☒

CONVERT 

BRUSH 


Parameter T :



CLEAR 

Download



GOOD LUCK 





# 6

## Problem & Solution

1. 사용자의 스케치의 다양성

Project

1. Manifold learning

2. Face-outline 불일치로 인한 학습 성능 저하  
3. 데이터 추가의 번거로움

Data

2. 눈,입 위치의 정규화(고정)  
3. Docker와 airflow을 이용한 자동화

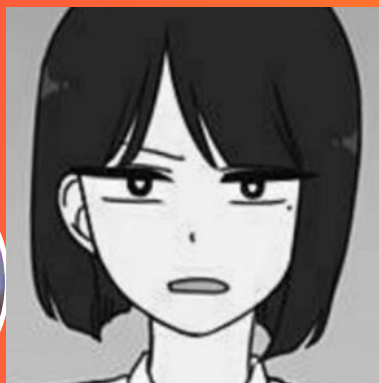
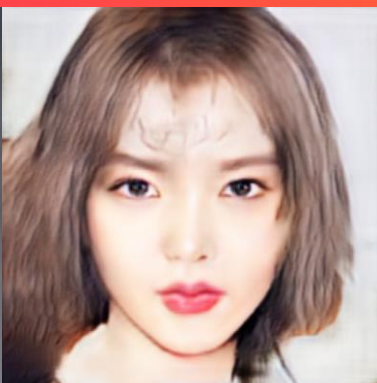
4. 입력 스케치 위치의 어긋남  
5. Stage1의 blurry한 결과물

Model

4. Stage 1에 augmentation 적용 (Resized crop)  
5. Stage 2에 augmentation 적용 (Blur, Cutout)

# 7 Further Work

- 다양한 목적  
: 남녀 얼굴 바꾸기,  
에니메이션 실사화
- 다양한 데이터  
: 몽타주로 용의자 그리기
- 최적화 및 경량화
- Ngnix를 통한 배포
- CI/CD를 통한 배포 자동화





THANK YOU





# Q&A - data

- 1. google api를 사용했는데 유료이지 않는가. 계속 활용할 수 있는 방법인가?
- => 우리는 우리의 자원내에서 최대의 효율을 낼 수 있는 방법을 생각했다. google api가 가장 정확하게 눈코입의 위치를 잡아냈다. 그리고 30만원의 free tier가 있어서 현재 사용중이다. 하지만 대안 또한 마련되어있다. RetinaNet을 사용한다면 아주 조금 눈코입 포인트가 틀어지지만 훈련에 사용할 수 있는 데이터가 생성된다.
- 2. 데이터관련 어려웠던 점.
- => 데이터 팀과 모델 팀간의 데이터에 관한 기준이 달라 조율하는 과정이 오래걸렸다. 하지만 많은 토의를 통해 최적의 데이터셋 필터링 규칙을 만들어냈다.
- 3. 업로드 서버에 해킹위험?
- => 파일 필터링을 통해 이미지 외의 데이터는 수신하지 않도록 함으로써 악성 프로그램이 다운되는걸 방지했다.
- 4. airflow는 무겁지 않은가 bash shell로도 충분히 될 것 같다.
- => bash shell과 crontab을 사용해 동일한 과정을 이미 만들었다. 하지만 기술을 보여주는 자리이기 때문에 우리팀이 다양한 것을 시도했다는 것을 보여주고 싶었다.
- 5. 효율성을 높이기 위해 한 시도와 효율성을 위해 앞으로 무엇을 더할 수 있나
- => 효율성을 높이기 위해 크게는 모든 데이터과정을 자동화하였다. 또한 각자 이미지를 서버에 업로드만 하면되기때문에 메일로보내고 압축푸는등의 불필요한 과정을 모두 스킵하였다.
- 추가적으로 효율성을 높이기 위해 시도해보고 있는것은 이미지를 전처리하는 도커에서 cpu나 gpu사용량이 최대를 사용하지 않는다는 것을 확인했다. 따라서 각각의 도커에 자원을 적게 주고 도커를 더 많이 생성해 이미지를 병렬적으로 처리함으로써 효율성을 높일 계획이다.

# Q&A – model\_stage1

- 1. k-nn 말고도 임베딩된 벡터를 조합하는 방법이 무엇이 있을까요? 이나 지금 모델에서 개선 할점 질문 들어왔을때 예상 답변
  - K-nn은 모든 데이터와 l2 distance를 계산해야 되기때문에  $O(N)$ 의 시간복잡도를 가진다. inference 단계중 KNN을 수행하는 시간이 가장 긴데, 추후에 모델 성능 향상을 위해 더 많은 데이터를 사용한다면 KNN 수행 때문에 inference 시간이 더욱 더 늘어 날것 같다. 따라서, 인퍼런스 수행 시간을 줄이려면 knn 연산을 GPU를 사용해서 병렬화를 하거나, ANNS(approximate nearest neighbor search)중 하나를 선택해서 수행 시간을 줄이는 방법을 택해야 될 것 같다.



# Q&A – model\_stage2

- 1. 모델 개선을 위해서 어떤 시도를 해볼 수 있을까요?
  - 다양한 GAN 시도
    - Modified CycleGAN
- 2. cycleGAN이 결과가 왜 상대적으로 품질이 낮은가?
  - 논문상에서 horse2zebra 실험을 한 것 처럼, 소수의 데이터로 적용할 때는 부분적인 생성또는 변화만 가능하다. 특히 데이터의 분포가 불안정할 경우 이미지를 제대로 생성하지 못한다. 우리의 목적은 전체적인 이미지를 생성해야하기 때문에 어려움이 있다고 생각한다. 또한, 128사이즈의 비슷한 목적의 pretrained model 이 있어 실험이 가능했지만, 512사이즈는 시간 대비 성능에 큰 변화가 없었으며 한정된 gpu에서 실험을 지속적으로 하기에는 한계가 있었다.



# Q&A – Serving

- 1.