PORTFOLIO
CONTENTS

01.	Image generation & Voice conversion	SOTA LDMs 및 LLM 모델을 사용하여 맞춤화 된 솔루션 제공
02.	3D Volume and Circumference Algorithm 개발	3D Points를 이용하여 부피 및 둘레를 계산하는 기능을 구현 진행
03.	3D Human Body Segmentation with DL 기획 및 개발	Fitted Bra Size 추천을 위해 가슴 부분을 정확하게 분할하는 모델 개발 진행
04.	Virtual Fitting Room App	Core 프로젝트로 Depth Sensor를 이용하여 3D object를 구성하여 고객에게 맞는 사이즈를 제공하는 3D scanning 파트 개발 및 구현 진행
05.	Object Tracking	Machine learning based LiDAR object tracking

최근에 진행한 **5개의 프로젝트 내용을 서술하였습니다.*

Project 1.

Image Generation & Voice Conversion

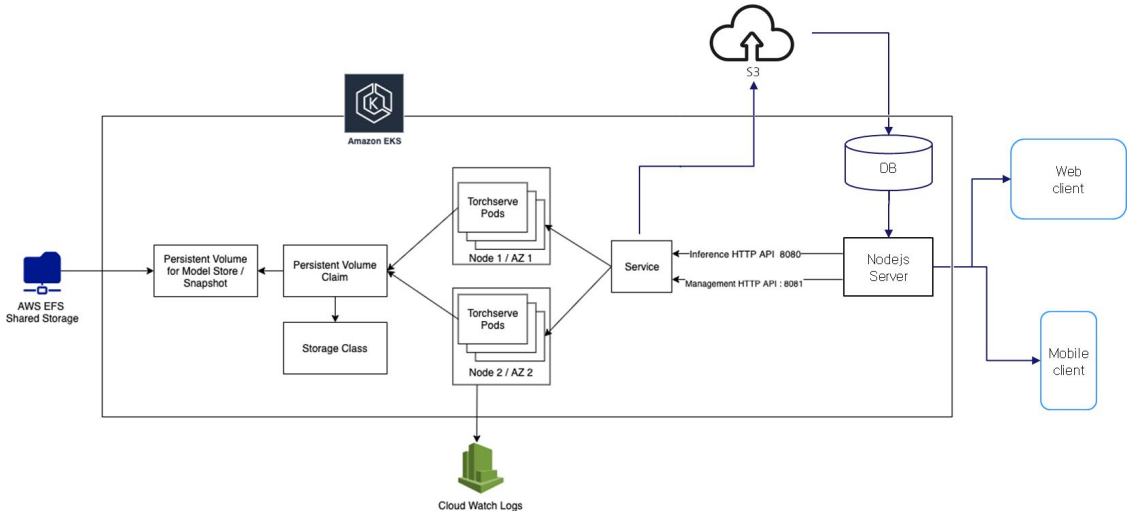
About project

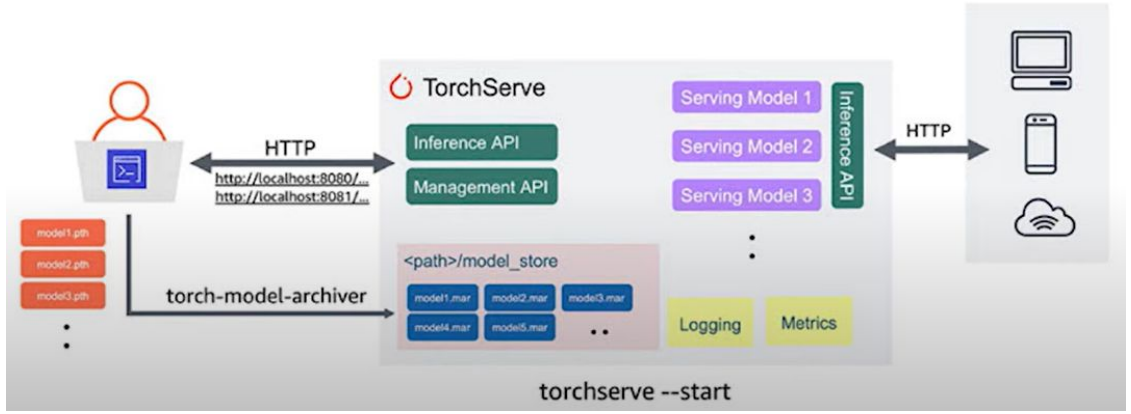
인디게임사에서, 평소 인력이 부족할 뿐 아니 한 사람이 여러 파트를 작업을 하다보니, 시간효율적 업무를 못할때가 많다. 인력 및 시간을 최소화하면서, 더 나은 퀄리티를 제공하는 24시간 생성ai SaaS를 상용화하여, 인련난에 어려움을 격는 스타트업 및 벤처기업에 솔루션을 제공하여 많은 제품을 개발 할 수 있도록 서비스 개발하는 것을 목표로 시작하여, Custom Game Assets 생성, 만화 캐릭터 생성 및 보이스 변환 개발을 진행하고 있습니다.

Project 1. Image generation & Voice conversion

Introduce project

작업 기간	2023. 06 ~ 현재
인력 구성(기여도)	기획 1명, FE 1명, BE 2명, AI 2명 (AI 기여도 80%)
프로젝트 목적	Custom Game Assets 생성 및 만화 캐릭터 생성
프로젝트 내용	LDMs 모델을 기반 및 사용자가 Noise layers weight를 조정가능하게하여 사용자 학습 데이터와 함께 맞춤형 이미생 생성 SaaS 개발
주요 업무 및 상세 역할	1) LDM Custom model 설계 2) 정량 학습 데이터셋 생성 3) BE&FE AI core 연결 파이라인 초안 기획
사용언어 및 개발 환경	Linux, Python, AWS, Pytorch
참고 자료	PPT 자료 참고





Original
Flat2DAnimerge

Add More Details - Detail Enhancer／Tweaker (细节调整) LoRA

prompt = "face focus, cute, masterpiece, best quality, 1girl, green hair, sweater, looking at viewer, upper body, beanie, outdoors, night, turtleneck"

negative_prompt = "lowres, bad anatomy, bad hands, text, error, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry"

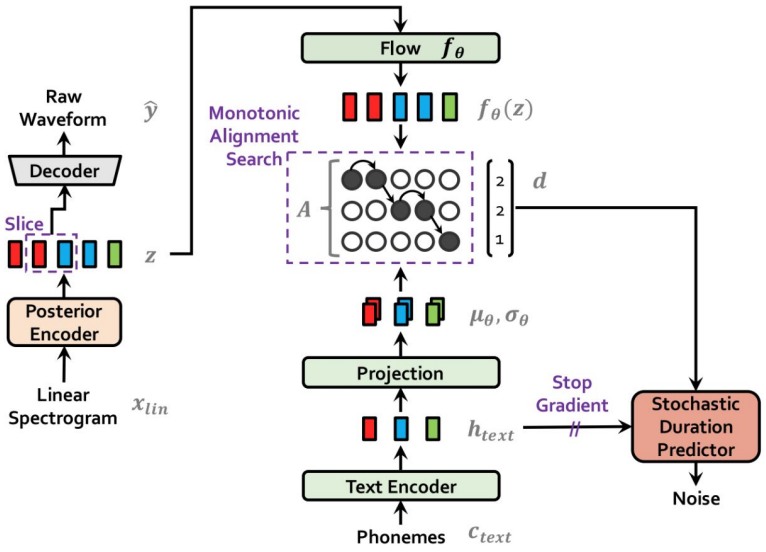
scale = 1

generator = torch.Generator().manual_seed(1337)

Sampler: Euler A

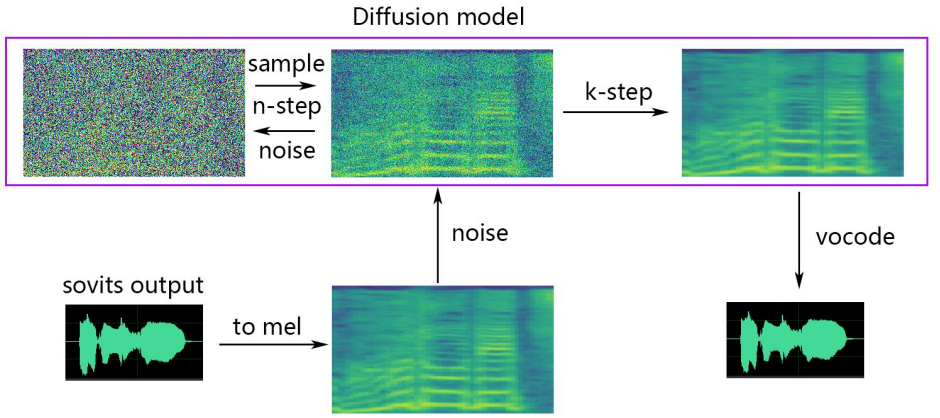
Introduce project

작업 기간	2023. 12 ~ 현재
인력 구성(기여도)	기획 1명, AI 1명 (AI 기여도 100%)
프로젝트 목적	게임 산업에 사용되는 나레이션 및 게임 효과음 생성
프로젝트 내용	LLM 및 LDM 모델을 사용하여, Any-to-One VC 모델 개발, 모델 재배포 및 맞춤 학습 가능한 MLOps 파이프라인 구성
주요 업무 및 상세 역할	1) LDM Custom model 설계 2) 정량 학습 데이터셋 생성 3) BE&FE AI core 연결 파이라인 구상
사용언어 및 개발 환경	Linux, Python, AWS, Pytorch
참고 자료	PPT 자료 참고



(a) Training procedure

(b) Inference procedure



Project 2.

3D Volume and Circumference Calculation Algorithm

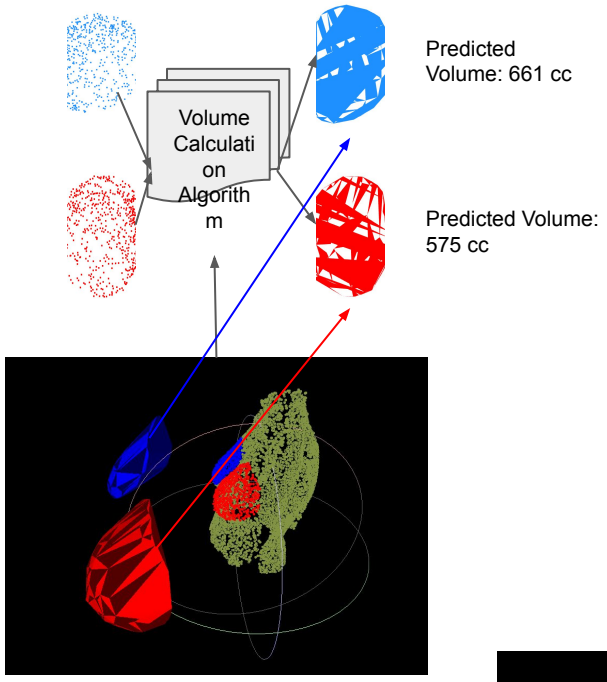
About project

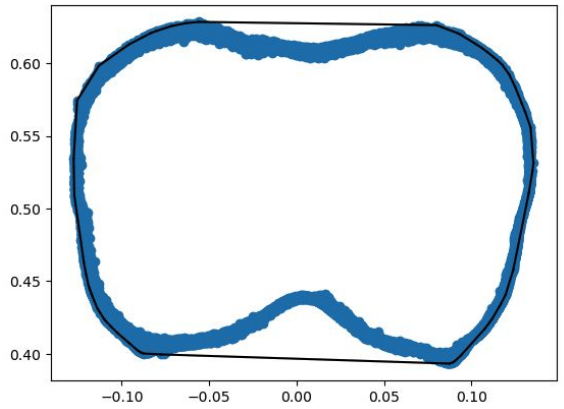
단순한 길이 계산을 넘어, 둘레 및 부피 측정 알고리즘을 구현하여 Virtual Fitting Room에 도입하여 장소 상관없이 자신에게 맞는 옷 사이즈를 추천받을 수 있는 플랫폼을 구축하는 것 입니다. 뿐만 아니라, 다양한 산업군에서도 스캔을 통해 사람이 측정하는 기준보다 더 정확하고 정밀하게 측정 할 수 있는 다양한 솔루션으로 확장 될 수 있습니다.

Project 2. 3D Volume and Circumference Calculation Algorithm

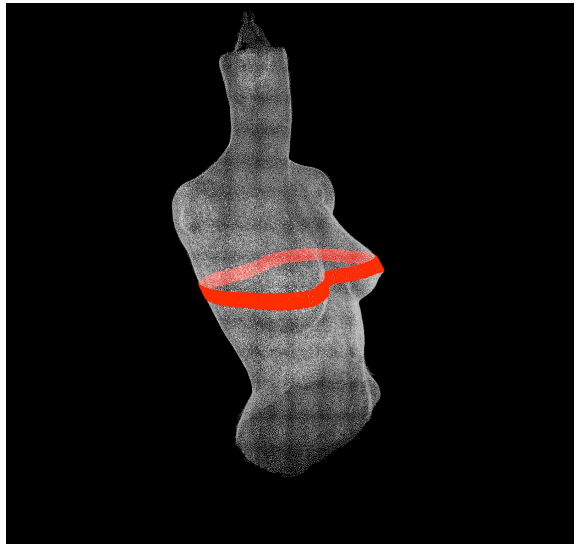
Introduce project

작업 기간	2022. 01 ~ 2023. 03 (3개월)
인력 구성(기여도)	CV1명 (CV 기여도 100%)
프로젝트 목적	정확한 3D 가슴 측정을 통하여 모양, 부피 및 둘레를 측정하여 기존에 사용되던 사이즈 측정방법을 대체하고 유니버셜 가슴 사이즈 확립
프로젝트 내용	segmentation을 통해 얻어진 가슴 3d points를 가지고 모양 분석 및 convex hull을 구성하여 3d 부피 재구성하여 부피측정하며 peak point method를 이용하여 윗 가슴 아랫가슴 둘레를 측정.
주요 업무 및 상세 역할	1) 부피계산 알고리즘 구현 2) Finding peak point 알고리즘 구현 3) 둘레 계산 알고리즘 구현
사용언어 및 개발 환경	Linux, Python, Open3D, Tensorflow
참고 자료	PPT 자료 참고





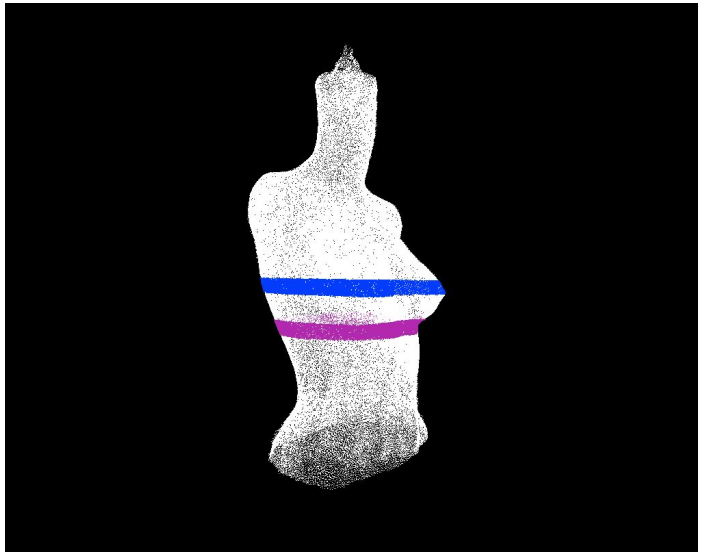
84.76273637556841cm

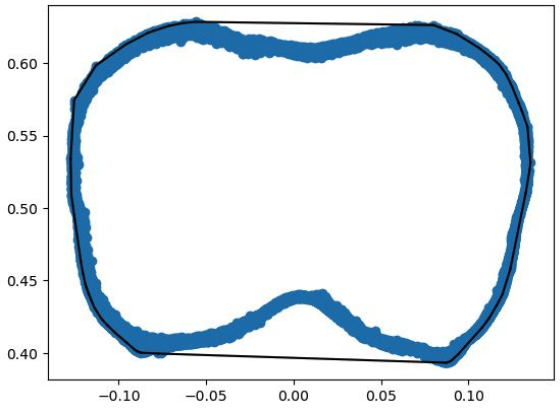


3D Volume and Circumference Calculation Algorithm

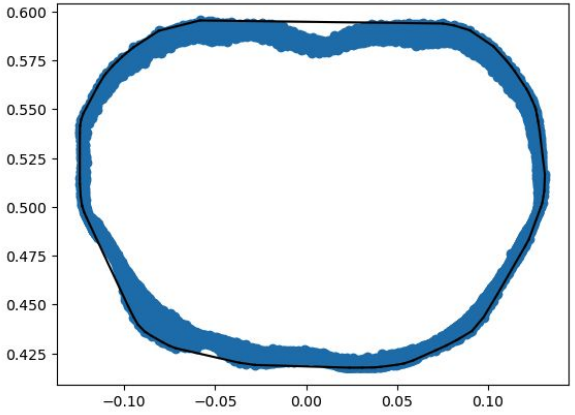
Main work

① 3D circumference calculation





84.76273637556841cm



58.7306304985048cm

```
def getUnderBurst():
    arr_pcd = np.asarray(pcd.points)
    side2d_points = []
    side3d_points = []

    for i in range(0, len(arr[hull.vertices])-1):
        sum += distance.euclidean(arr[hull.vertices][i], arr[hull.vertices][i+1])
```

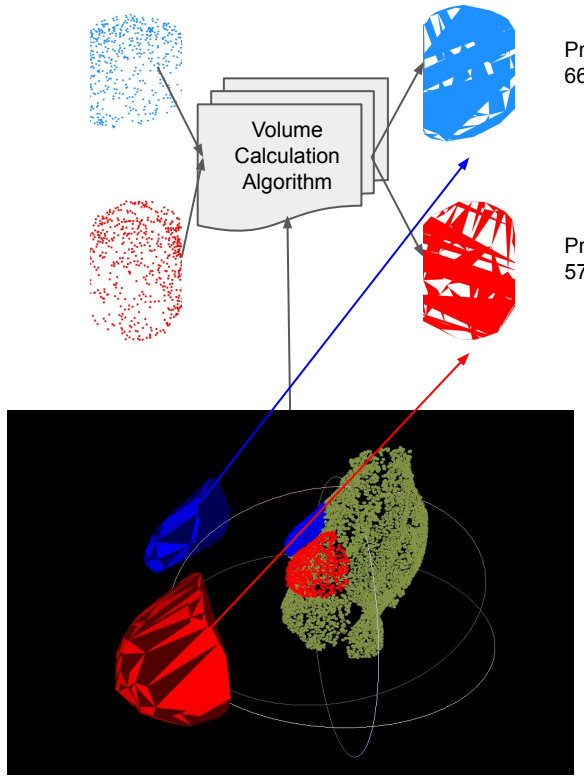
```
def getUpperBurst():
    arr_pcd = np.asarray(pcd.points)
    xy_point = []
    xyz_point = []

    for i in range(0, len(arr[hull.vertices])-1):
        sum += distance.euclidean(arr[hull.vertices][i], arr[hull.vertices][i+1])
```

3D Volume and Circumference Calculation Algorithm

Main work

②3D Volume Calculation



Predicted Volume:
661 cc

Predicted Volume:
575 cc

```
def l_breastCalc():
    pcd = o3d.io.read_point_cloud('/home/nuonvision/result/real_model/outputleft1.ply')
    alpha = 17

def r_breastCalc():
    pcd = o3d.io.read_point_cloud('/home/nuonvision/result/real_model/outputright1.ply')
    alpha = 17
```

```
alpha=17.000
left: 248.80734863565357 cm3
alpha=17.000
right: 316.69966883570146 cm3
(segNet) nuonvision@NuVi:~/H-SegNet$
```

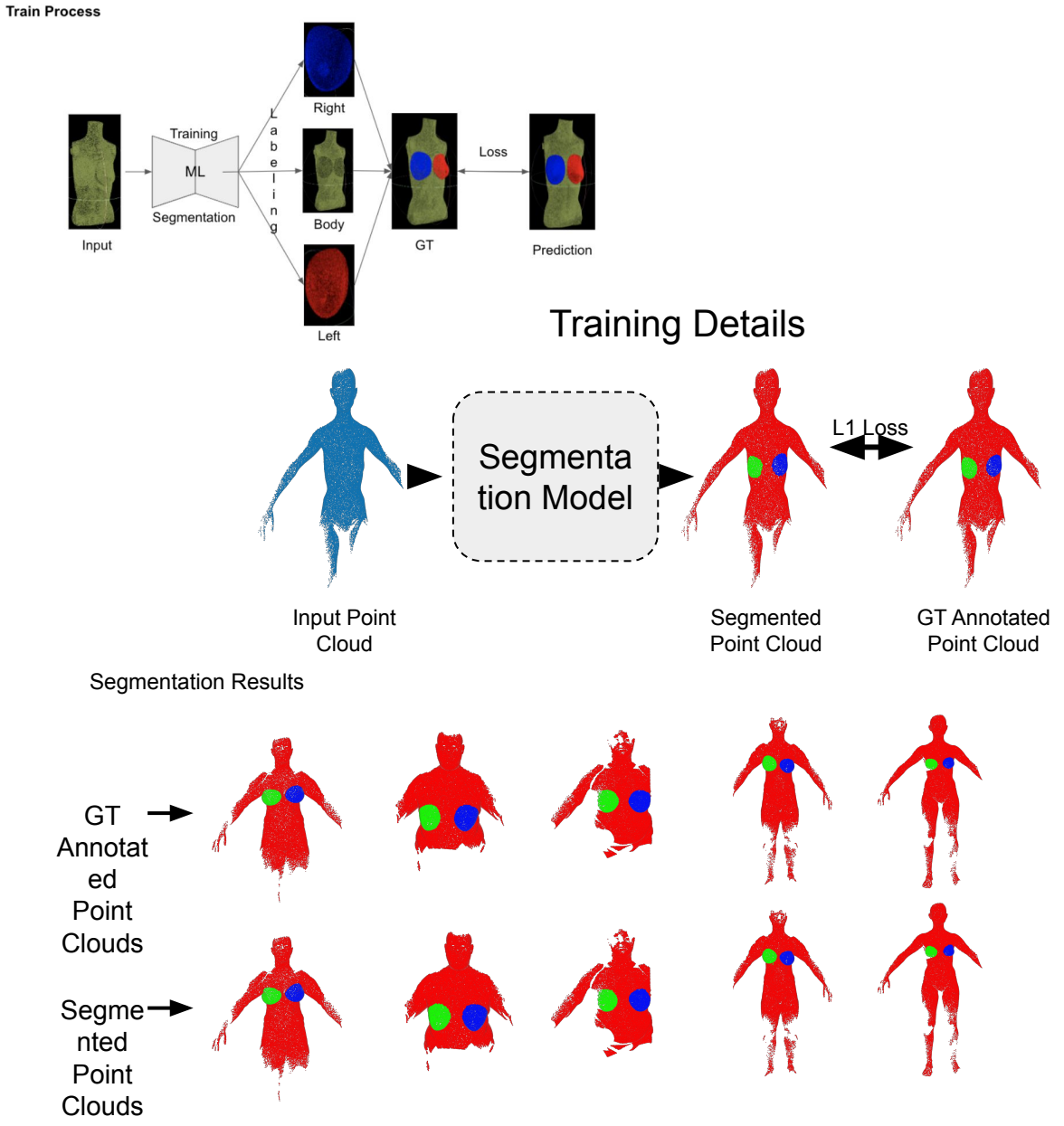
Project 3.

3D Human Body Segmentation with DL 기획 및 개발

About project

Virtual Fitting Room을 통해 장소 상관없이 자신에게 맞는 옷 사이즈를 추천받을 수 있는 플랫폼을 구축하여, 소비자에게 맞는 브랜드별 옷 사이즈를 매칭 및 추천 해주기 위해 먼저 사람 신체를 부위별로(머리, 목, 팔, 다리, 가슴, 몸통 등) 제대로 분할하는 기능을 개발하는 것 입니다.

3D Human Body Segmentation with DL



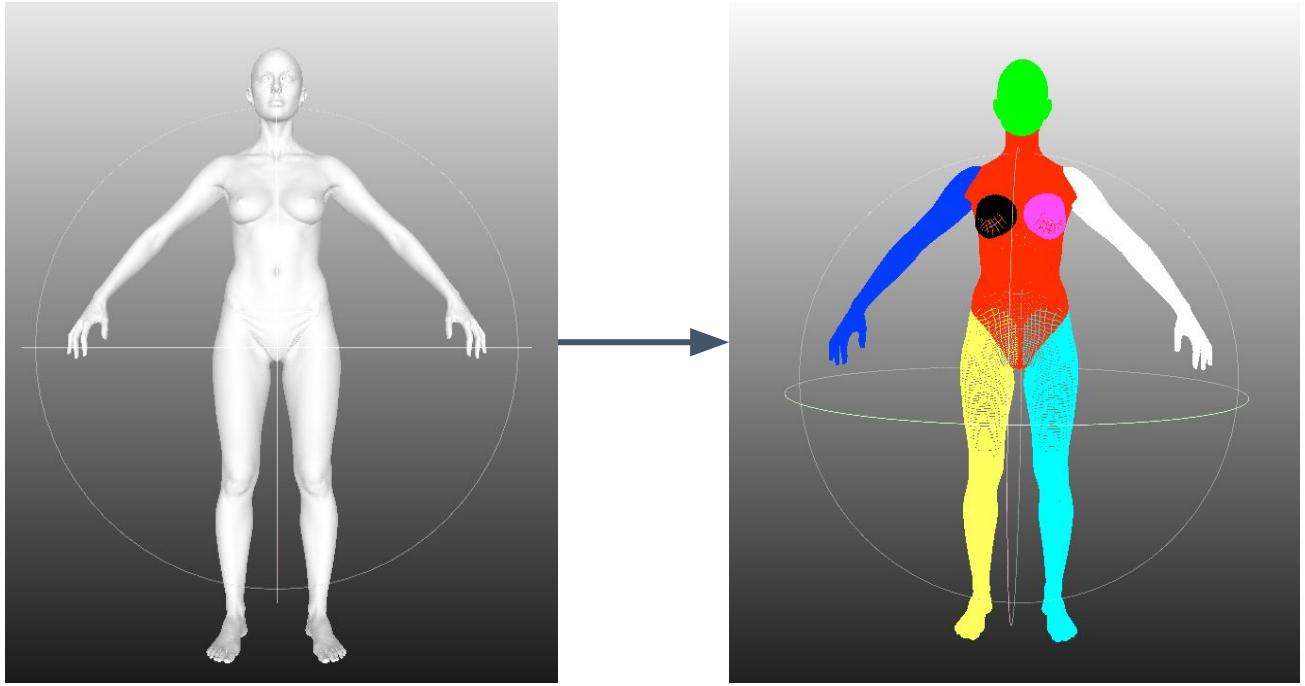
Introduce project

작업 기간	2021. 11 ~ 2022. 06 (8개월)
인력 구성(기여도)	CV3명 (기여도 50%)
프로젝트 목적	원시적인 가슴사이즈 측정을 현대에 정확하고 고객이 매장을 방문하지 않고 직접 집에서 자유롭게 핸드폰으로 3D Model을 만들어 가슴형태와 가슴크기를 분석하여 고객에게 맞는 브래지어 사이즈를 추천
프로젝트 내용	사용자가 사이즈 촬영을 시작할 경우, Depth Sensor로 Captured RGB-D frames을 registration, 가슴 측정을 위한 사용자의 3D model를 반환
주요 업무 및 상세 역할	1) H-SegNet 구성 2) Data Labeling 및 automatic tool about fixed topology 구현 3) Optimization DL model
사용언어 및 개발 환경	Open3D, Python, Linux, GitLab, TensorFlow
참고 자료	PPT 자료 참고

3D Human Body Segmentation with DL

Main work

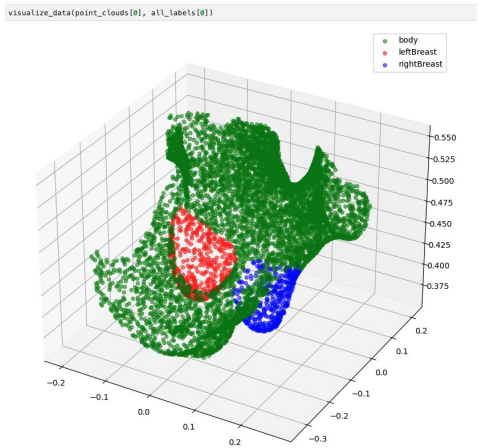
①Unified Data Labeling



```
def labeling(i):
    pcd = o3d.io.read_point_cloud('../Downloads/morpha
    xyz = np.asarray(pcd.points)

    f1 = open('../Downloads/bodyIdx/body.txt')
    f2 = open('../Downloads/bodyIdx/head.txt')
    f3 = open('../Downloads/bodyIdx/leftArm.txt')
    f4 = open('../Downloads/bodyIdx/leftBreast.txt')
    f5 = open('../Downloads/bodyIdx/leftLeg.txt')
    f6 = open('../Downloads/bodyIdx/rightArm.txt')
    f7 = open('../Downloads/bodyIdx/rightBreast.txt')
    # f7 = open('../Downloads/rightBreast.txt')
    f8 = open('../Downloads/bodyIdx/rightLeg.txt')
```

Main work ②Segmentation Training



- **기능 소개**

- Labeled dataset을 가지고 segmentation training 실행

- **작업 내용**

- Human Body Segmentation

  > Architect DL model for Human Segmentation

  > Optimization for best performance with layers and hyper-parametres

```
with open("/home/nuonvision/data_set/selectedMorph/metadata.json") as json_file:
    metadata = json.load(json_file)

points_dir = "/home/nuonvision/data_set/selectedMorph/{}/points".format(
    metadata["baseDistance"]["directory"]
)
labels_dir = "/home/nuonvision/data_set/selectedMorph/{}/point_labels".format(
    metadata["baseDistance"]["directory"]
)

LABELS = metadata["baseDistance"]["lables"]
COLORS = metadata["baseDistance"]["colors"]
```

```
training_step_size = total_training_examples // BATCH_SIZE
total_training_steps = training_step_size * EPOCHS
print(f"Total training steps: {total_training_steps}.")

lr_schedule = keras.optimizers.schedules.PiecewiseConstantDecay(
    boundaries=[training_step_size * 15, training_step_size * 15],
    values=[INITIAL_LR, INITIAL_LR * 0.5, INITIAL_LR * 0.25],
)

steps = tf.range(total_training_steps, dtype=tf.int32)
lrs = [lr_schedule(step) for step in steps]

plt.plot(lrs)
plt.xlabel("Steps")
plt.ylabel("Learning Rate")
plt.show()
```

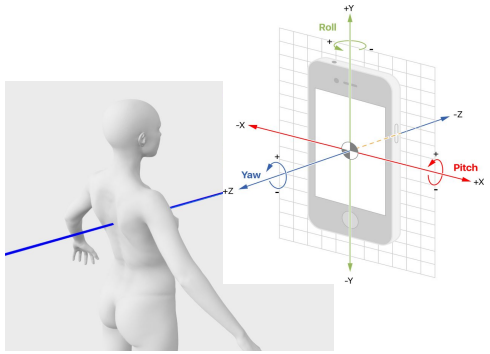
Project 4

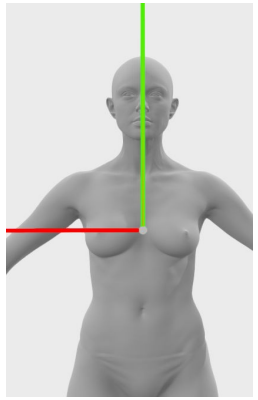
Virtual Fitting Room App

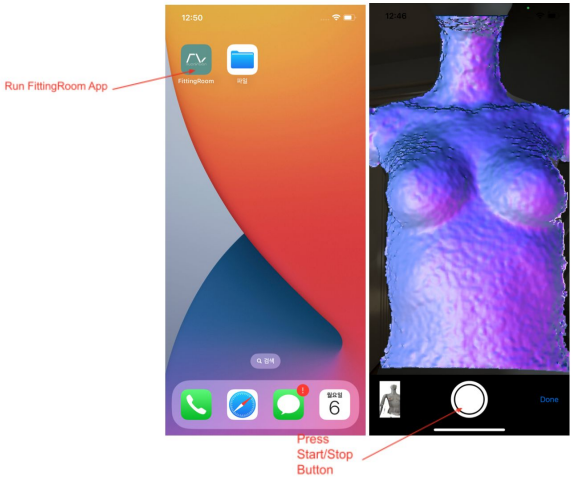
About project

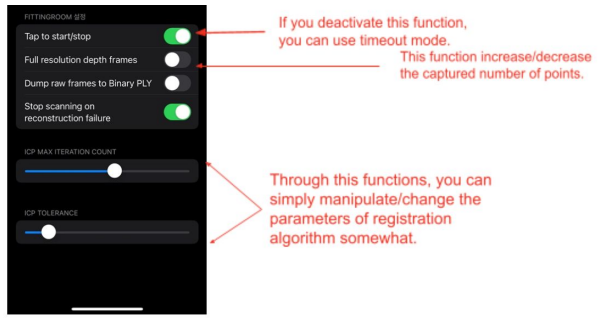
Covid 및 제약적인 사이즈 측정 환경의 문제를 해결하여,
사용자의 공간적 제약을 줄이고, 정확한 Sensor 데이터를 통해
원시적이었던 의류 사이즈 측정 방식을 대체하여 현대적이고,
정확한 사이즈를 통해 맞춤 의류 사이즈를 고객에게 전달해 줄 수
있는 플랫폼을 구축하는 것입니다.

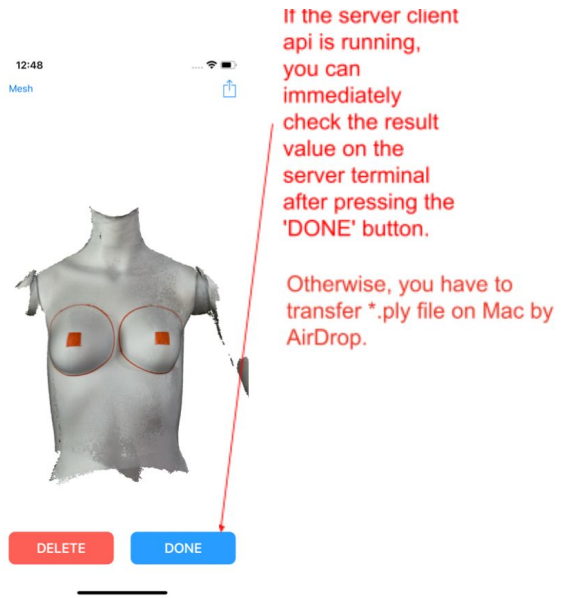
Project 4. Virtual Fitting Room App 개발







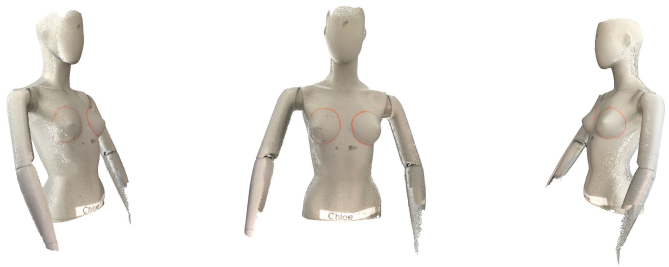




Introduce project

작업 기간	2021. 11 ~ 2022. 06 (8개월)
인력 구성(기여도)	BE 1명, CV1명, FE1명 (CV 기여도 100%)
프로젝트 목적	원시적인 가슴사이즈 측정을 현대에 정확하고 고객이 매장을 방문하지 않고 직접 집에서 자유롭게 핸드폰으로 3D Model을 만들어 가슴형태와 가슴크기를 분석하여 고객에게 맞는 브래지어 사이즈를 추천
프로젝트 내용	사용자가 사이즈 촬영을 시작할 경우, Depth Sensor로 Captured RGB-D frames을 registration, 가슴 측정을 위한 사용자의 3D model를 반환
주요 업무 및 상세 역할	1) RGB-D registration 알고리즘 구현 2) Scanning UI구성 3) Tab Navigation 구성
사용언어 및 개발 환경	Open3D, Swift, Python, Xcode, MacOS, Linux, GitLab
참고 자료	PPT 자료 참고

Main work 3D Registration 기능 구현



- **기능 소개**

- In real-time, 3D human body scanning 통해, 3D model 구성

- **작업 내용**

- Human Body Reconstruction
  > Customized ICP
  > Assimilating color, depth and IMU data.

```
#import <AVKit/AVKit.h>
#import <CoreMotion/CoreMotion.h>
#import <CoreVideo/CoreVideo.h>
#import <Foundation/Foundation.h>
#import <SceneKit/SceneKit.h>

NS_ASSUME_NONNULL_BEGIN

@class SCPointCloud;

@interface BPLYDepthDataAccumulator : NSObject

- (instancetype)init;

- (void)accumulateColorBuffer:(CVPixelBufferRef)colorBuffer
                    colorTime:(CMTime)colorTime
                  depthBuffer:(CVPixelBufferRef)depthBuffer
                    depthTime:(CMTime)depthTime
              calibrationData:(AVCameraCalibrationData *)calibrationData
NS_SWIFT_NAME(accumulate(colorBuffer:colorTime:depthBuffer:depthTime:calibrationData:));

- (void)accumulatePointCloud:(SCPointCloud *)pointCloud;
```

Project 5.

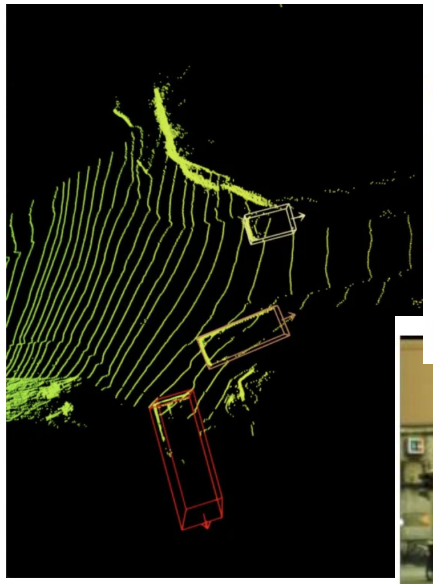
Object Tracking

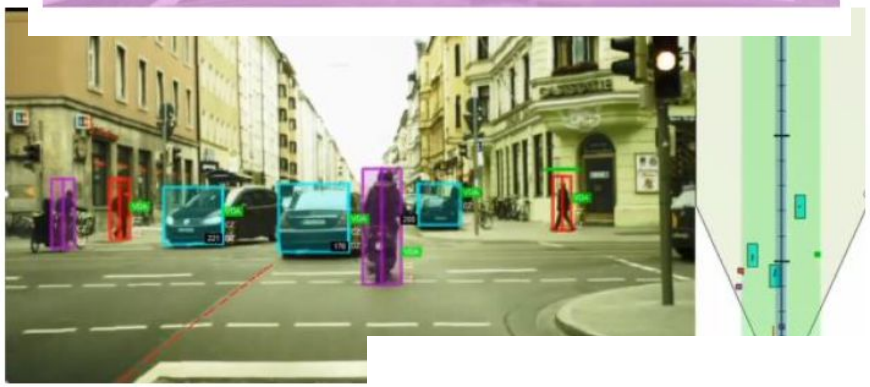
About project

This project follows the purpose of veoneer that create trust in mobility. Improving the object detection, provide a Tier-1 hardware supplier and system integrator focused on delivering innovative, best-in-class products and solutions.

Introduce project

Work period	2019. 10 ~ 2020. 09 (11 months)
Manpower composition (contribution)	BE 5 ppl, FE 3 ppl, ML eng 10 ppl, (ML contribution 2%)
Project purpose	This project is to serve the highly accurate object detection & tracking that create trust in ADAS.1
Content of the project	It is to implement accurate detection with ML about data received by camera and LiDAR sensor for reliable autonomous driving Lv5 stage.
Main Duties and Detailed Roles	- Researching the literature to gain a basic understanding of ML/DL state-of-the-art approaches - Assist the development and implementation of ML/DL-based systems for detecting and tracking objects in LiDAR point cloud data.
Language and development environment	Linux, Python, Open3D, Tensorflow





Main work ① Researching latest papers

- **Work Detail**
  - Review papers in a detail
  - Understanding recent object detection approaches
  - Support implementation/debugging of person detection error rate optimization

MMDetection: Open MMLab Detection Toolbox and Benchr

Kai Chen[1] Jiaqi Wang[1*] Jiangmiao Pang[2*] Yuhang Cao[1] Yu Xiong[1] X
Shuyang Sun[3] Wansen Feng[4] Ziwei Liu[1] Jiarui Xu[5] Zheng Zhang[6]
Chenchen Zhu[8] Tianheng Cheng[9] Qijie Zhao[10] Buyu Li[1] Xin Lu[4] Rui Zh
Jifeng Dai[6] Jingdong Wang[6] Jianping Shi[4] Wanli Ouyang[3] Chen Change Lo

[1]The Chinese University of Hong Kong [2]Zhejiang University [3]The University of Sydney [4]SenseT
[5]Hong Kong University of Science and Technology [6]Microsoft Research Asia [7]Beijing Institute of
[8]Nanjing University [9]Huazhong University of Science and Technology [10]Peking Univers
[11]Sun Yat-sen University [12]Northeastern University [13]Nanyang Technological Universit

Abstract

We present MMDetection, an object detection toolbox that contains a rich set of object detection and instance segmentation methods as well as related components and modules. The toolbox started from a codebase of MMDet team who won the detection track of COCO Challenge 2018. It

base with PyTorch [24].

Major features of MMDetection are:

sign. We decompose the detection fran
ent components and one can easily cons
object detection framework by combin
ules. (2) **Support of multiple framewor**
toolbox supports popular and contempoa

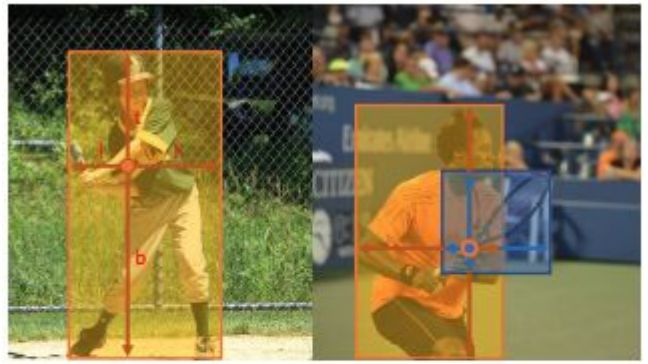


Figure 1 – As shown in the left image, FCOS works by predicting a 4D vector (l, t, r, b) encoding the location of a bounding box at each foreground pixel (supervised by ground-truth bounding box information during training). The right plot shows
hen a location residing in multiple bounding boxes, it
ambiguous in terms of which bounding box this location
regress.

Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick

Facebook AI Research (FAIR)

Abstract

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel *with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition. Code has been made available at: https://github.com/facebookresearch/Detectron.*

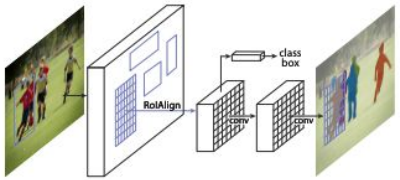
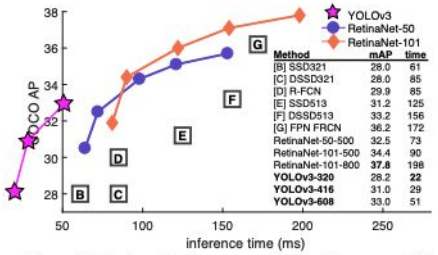


Figure 1. The **Mask R-CNN** framework for instance segmentation.

segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.[1] Given this, one might expect a complex method is required to achieve good results. However, we show that a surprisingly simple, flexible, and fast system can surpass prior state-of-the-art instance segmentation results.

Our method, called *Mask R-CNN*, extends Faster R-CNN [36] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in *parallel* with the existing branch for classification and bounding box regression (Figure 1). The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to-

ing



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

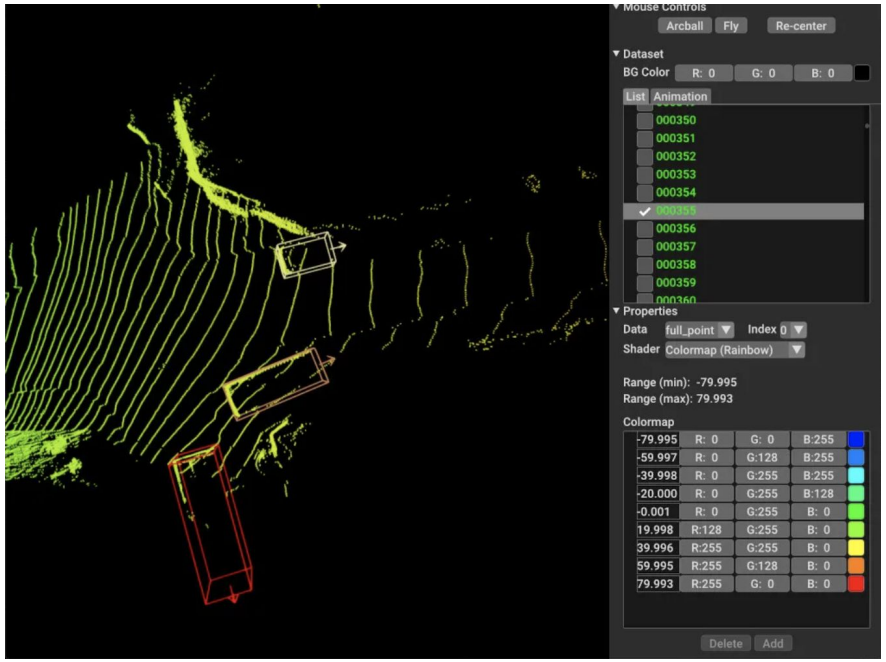
Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

2.1. Bounding Box Prediction

Following YOLO9000 our system predicts bounding

Main work

② Debugging and Performance check of Object Detection Model




```python
class ObjectDetection(BasePipeline):
    """Pipeline for object detection."""

    def __init__(self,
                 model,
                 dataset=None,
                 name='ObjectDetection',
                 main_log_dir='./logs/',
                 device='cuda',
                 split='train',
                 **kwargs):
        super().__init__(model=model,
                         dataset=dataset,
                         name=name,
                         main_log_dir=main_log_dir,
                         device=device,
                         split=split,
                         **kwargs)

    def run_inference(self, data):
        """Run inference on given data.
```

End of Document