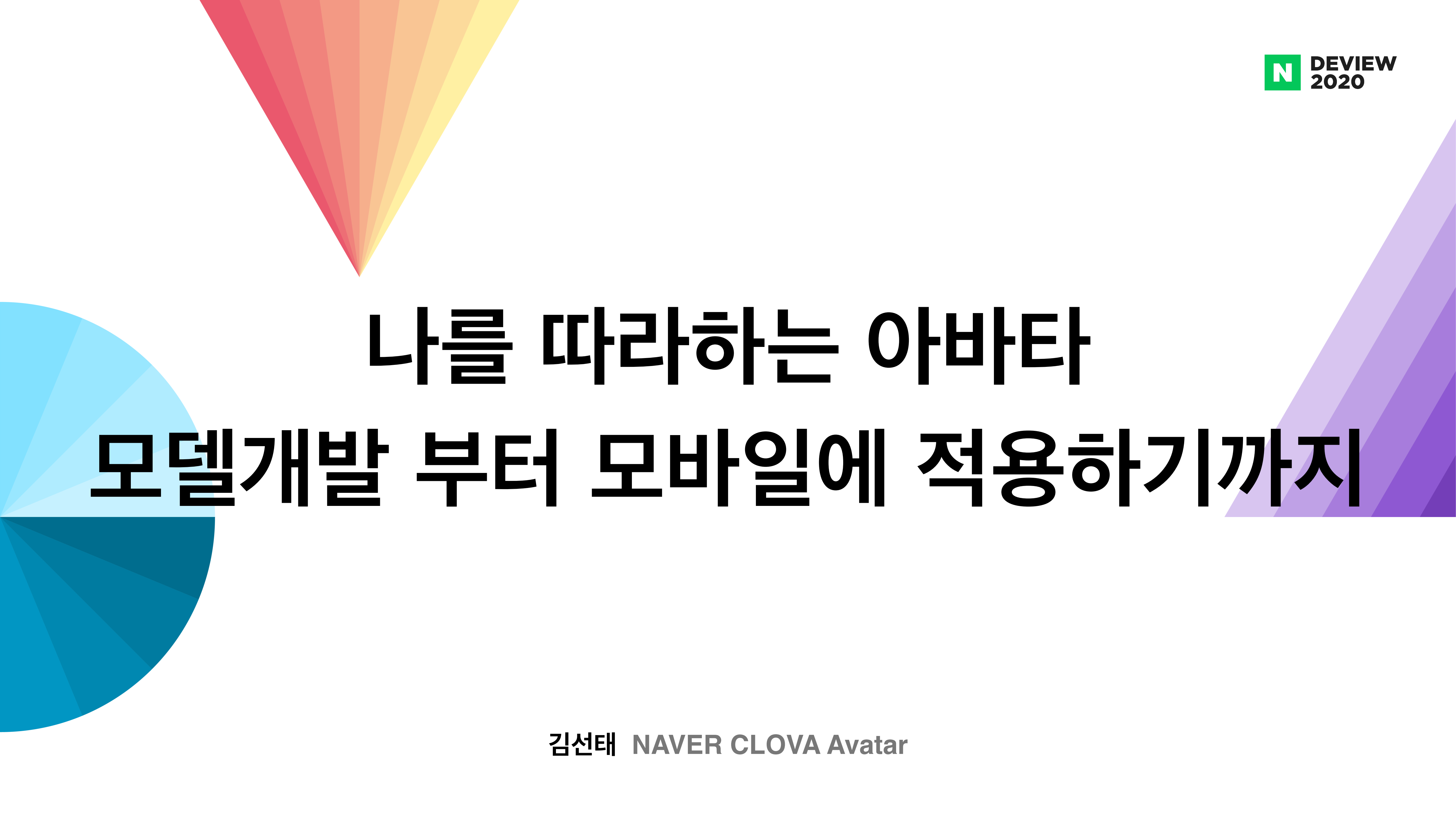




나를 따라하는 아바타 모델개발 부터 모바일에 적용하기까지

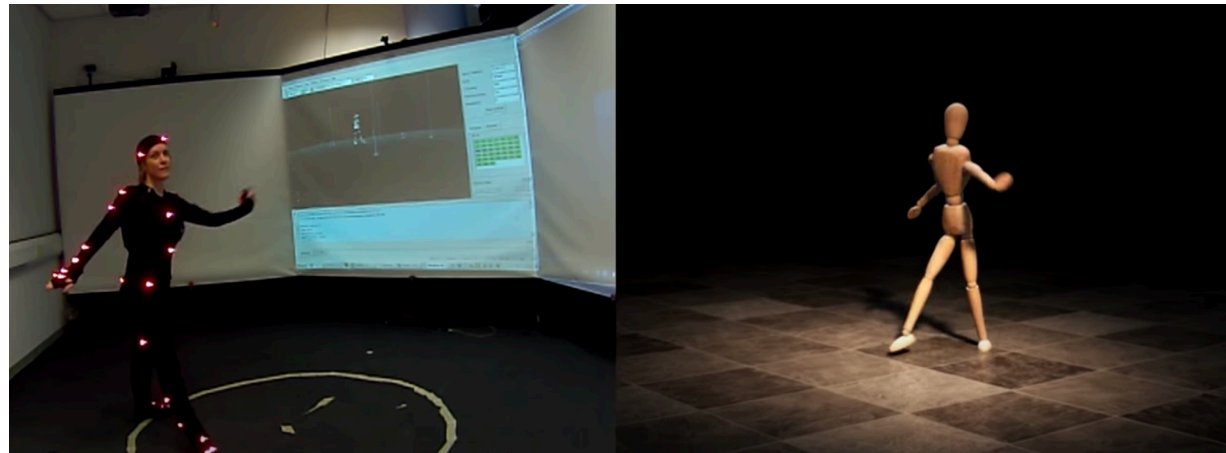
김선태 NAVER CLOVA Avatar

CONTENTS

- 1. Overview**
- 2. Failure Stories**
- 3. Lightweight 3D Human Pose Estimation**
- 4. SDK Development**
- 5. Mobile Application**
- 6. Optimization for Service**
- 7. QA Results**

1. Overview

지금 우리가 하고자 하는 것은?



Motion Capture 장비를 이용한 방식 (Sensor O)

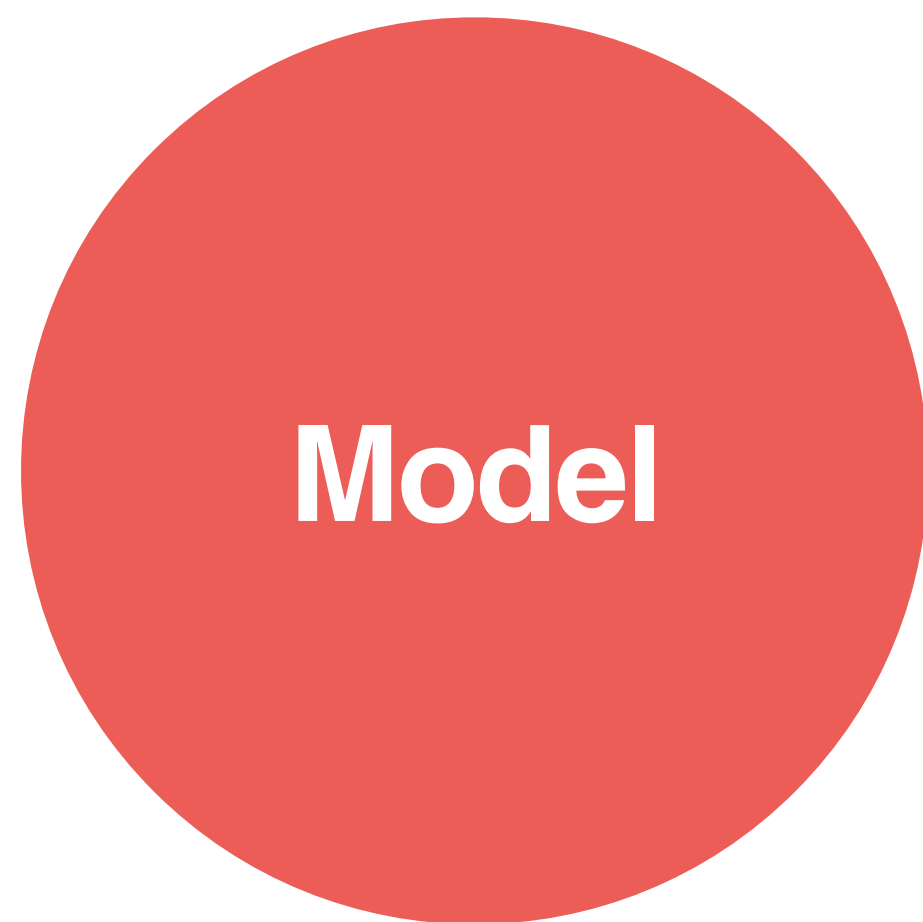


Deep Learning을 이용한 방식 (Sensor X)

- 사람의 3차원 자세 인식 (3D Pose Estimation)
- 실시간으로 나를 따라하는 3D 아바타 (Retargeting)

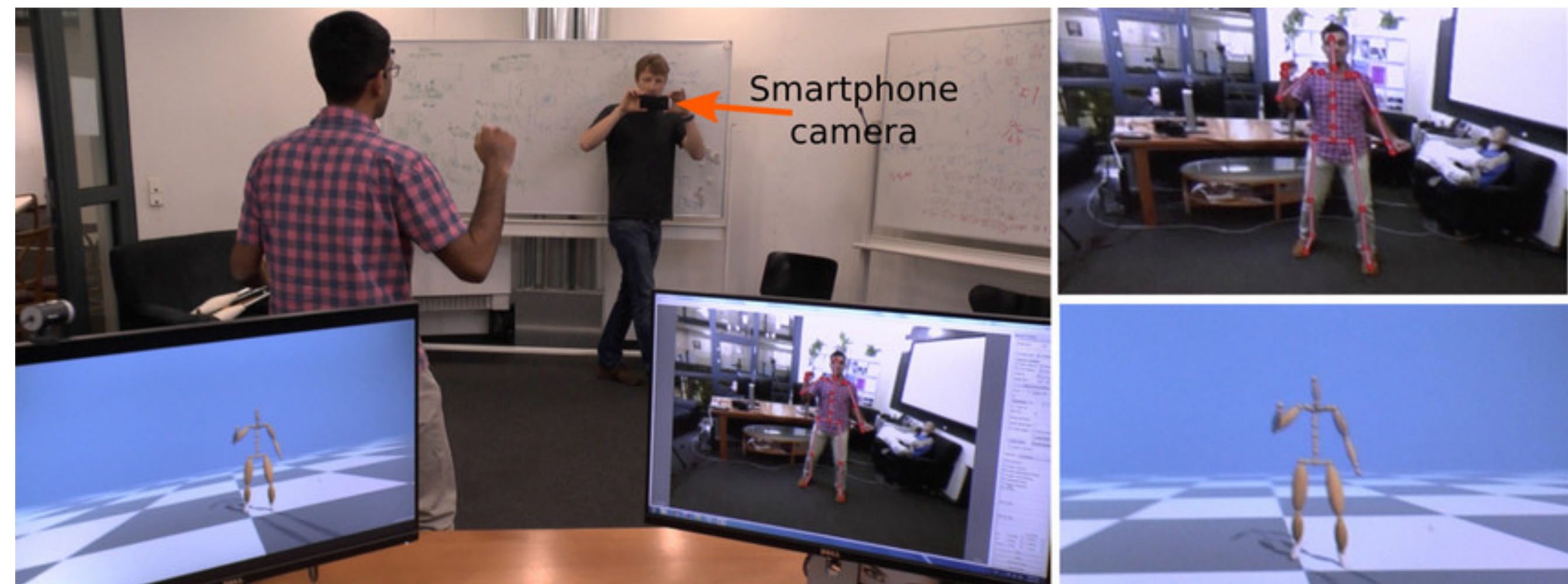
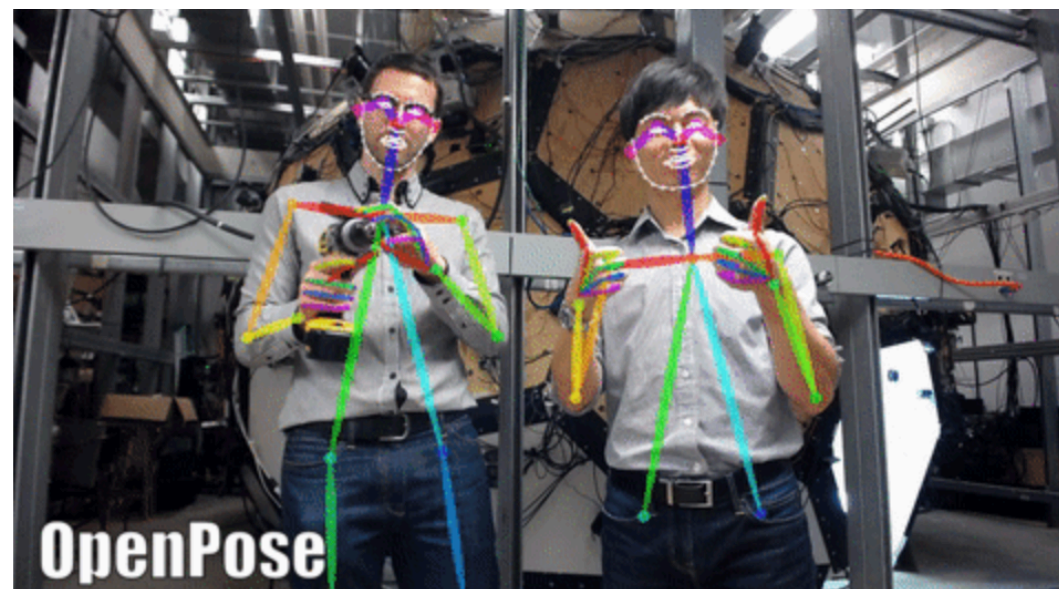
Components

- 사람의 자세를 3차원으로 예측할 수 있는 “Model”
- Model을 추론하고 결과값을 전달해줄 수 있는 “SDK”
- 추론 결과를 Visualization 해줄 수 있는 “Application”



Model (Deep Learning)

RGB 이미지 1장으로 사람의 자세를 3차원으로 예측하는 모델



<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

<https://github.com/tensorflow/tfjs-models/tree/master/posenet>

<http://gvv.mpi-inf.mpg.de/projects/VNect>

SDK (Inference Framework)

Input → Pre-process → Inference → Post-process → Output

Human Pose Engine

Pre-process

CoreML (iOS)

TFLite (Android)

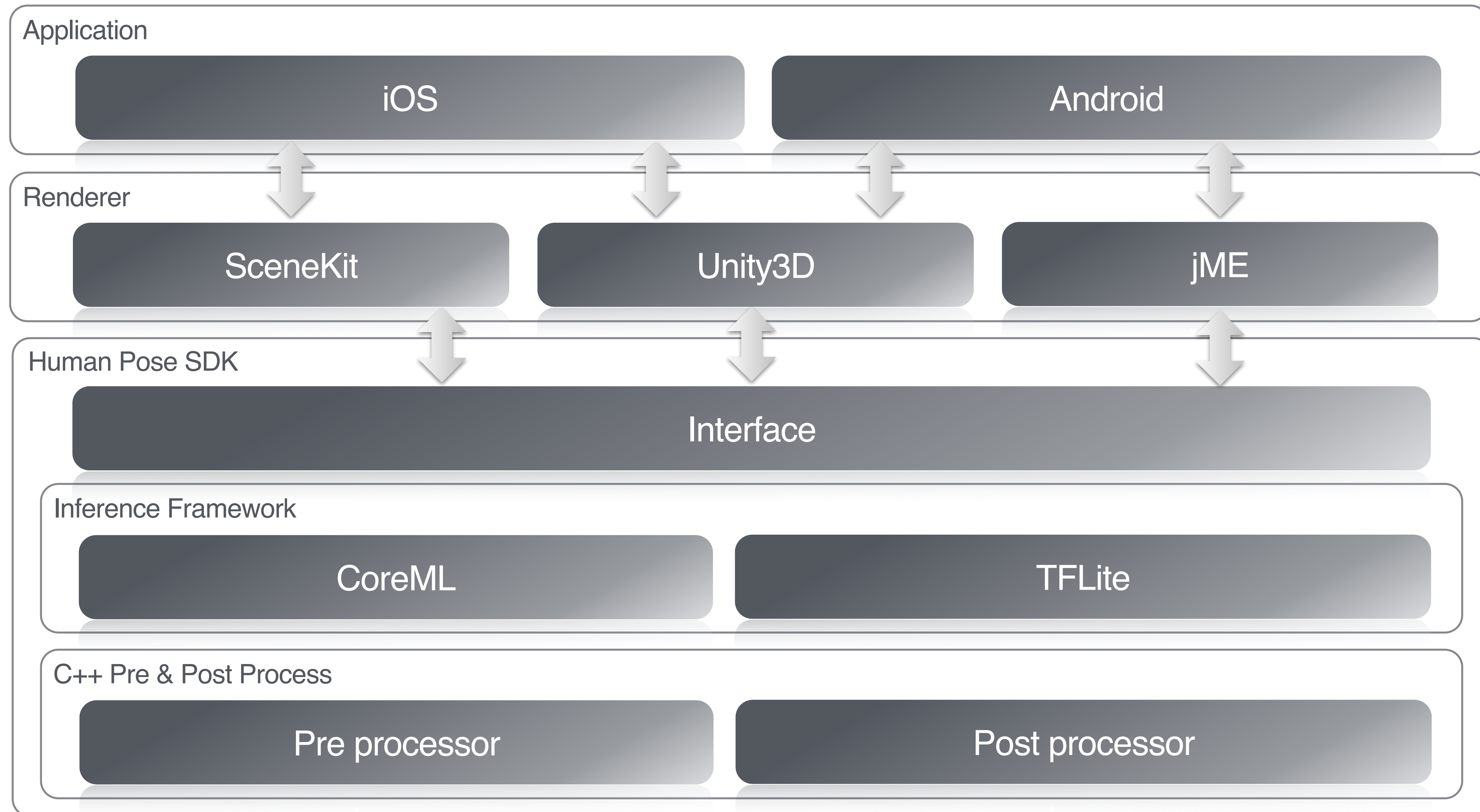
Post-process

Application (Graphics)

- 추론된 결과를 3D 캐릭터에 적용하여 애니메이션해주고 멋지게 연출
- 렌더링 및 애니메이션 리타게팅(Retargeting)

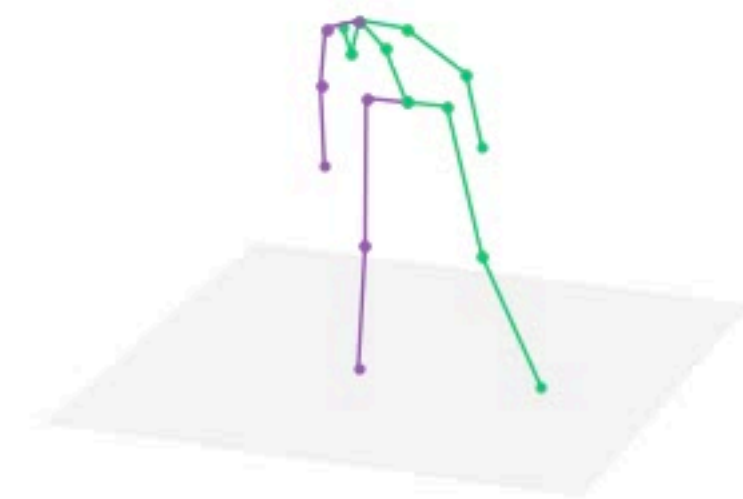
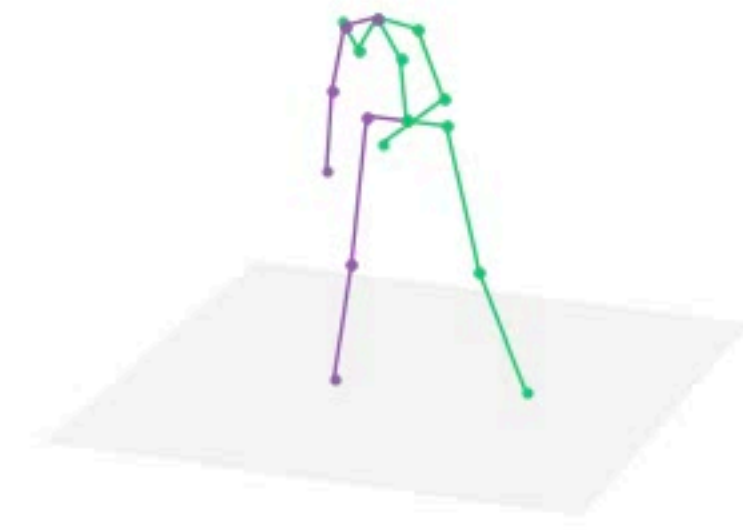


Architecture



2. Failure Stories

삽질 첫번째 (January, 2019)



좀비

팔, 다리 방향 안맞고..

고개는 왜 계속 숙여!!

삼질 두번째 (February, 2019)

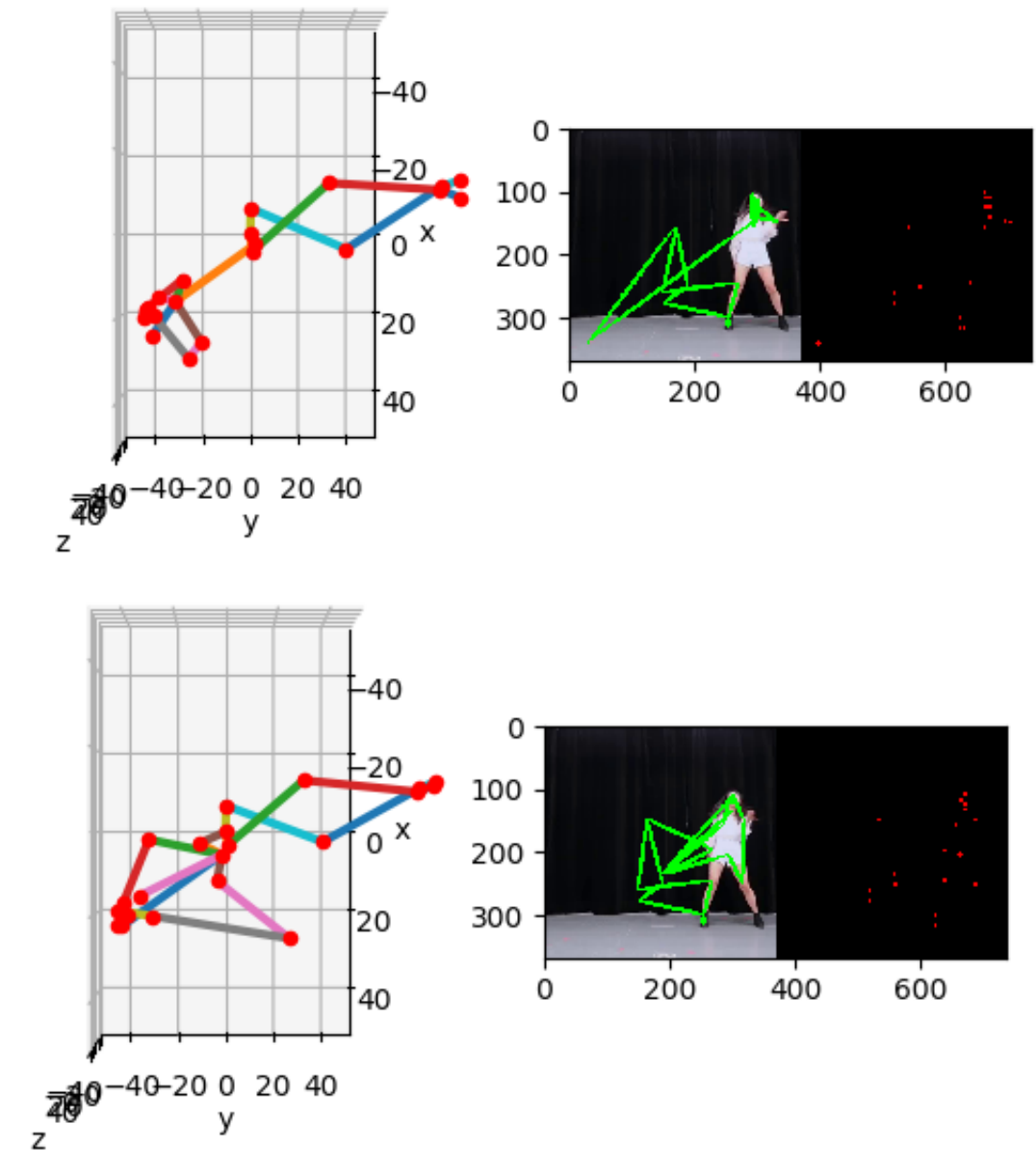
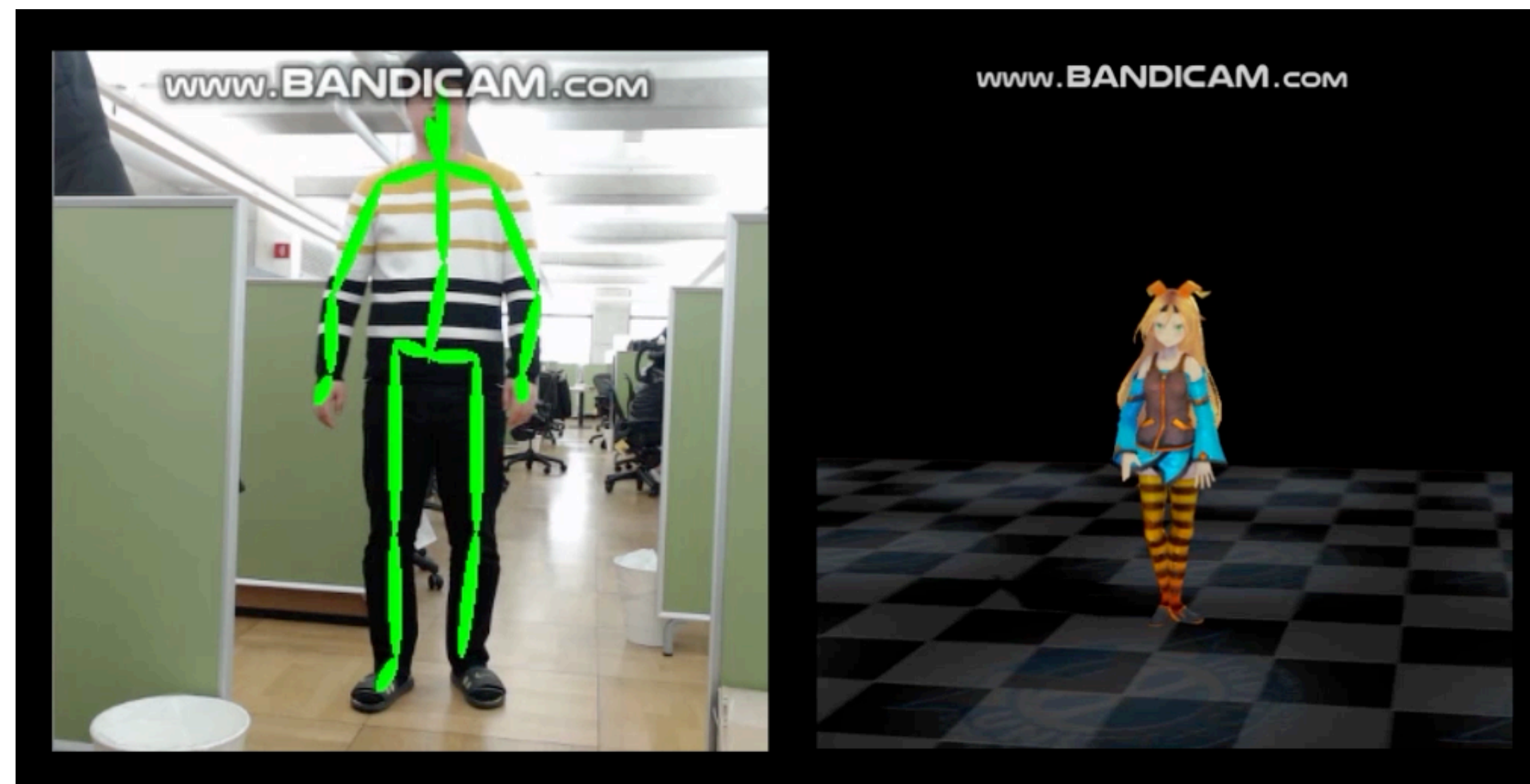


**Multi-scale Character에서
리타게팅이 가능하다는 것은
증명했지만,

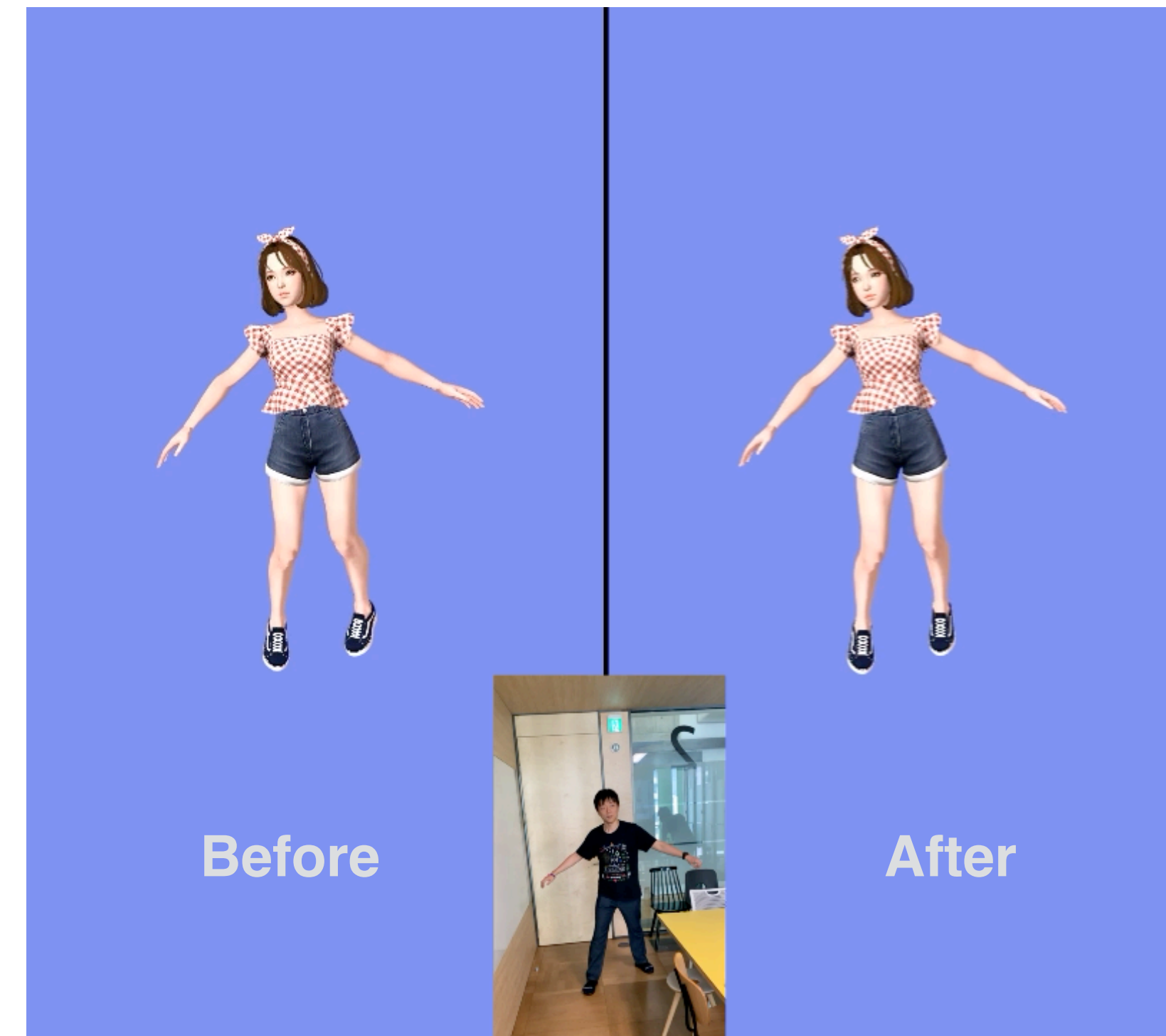
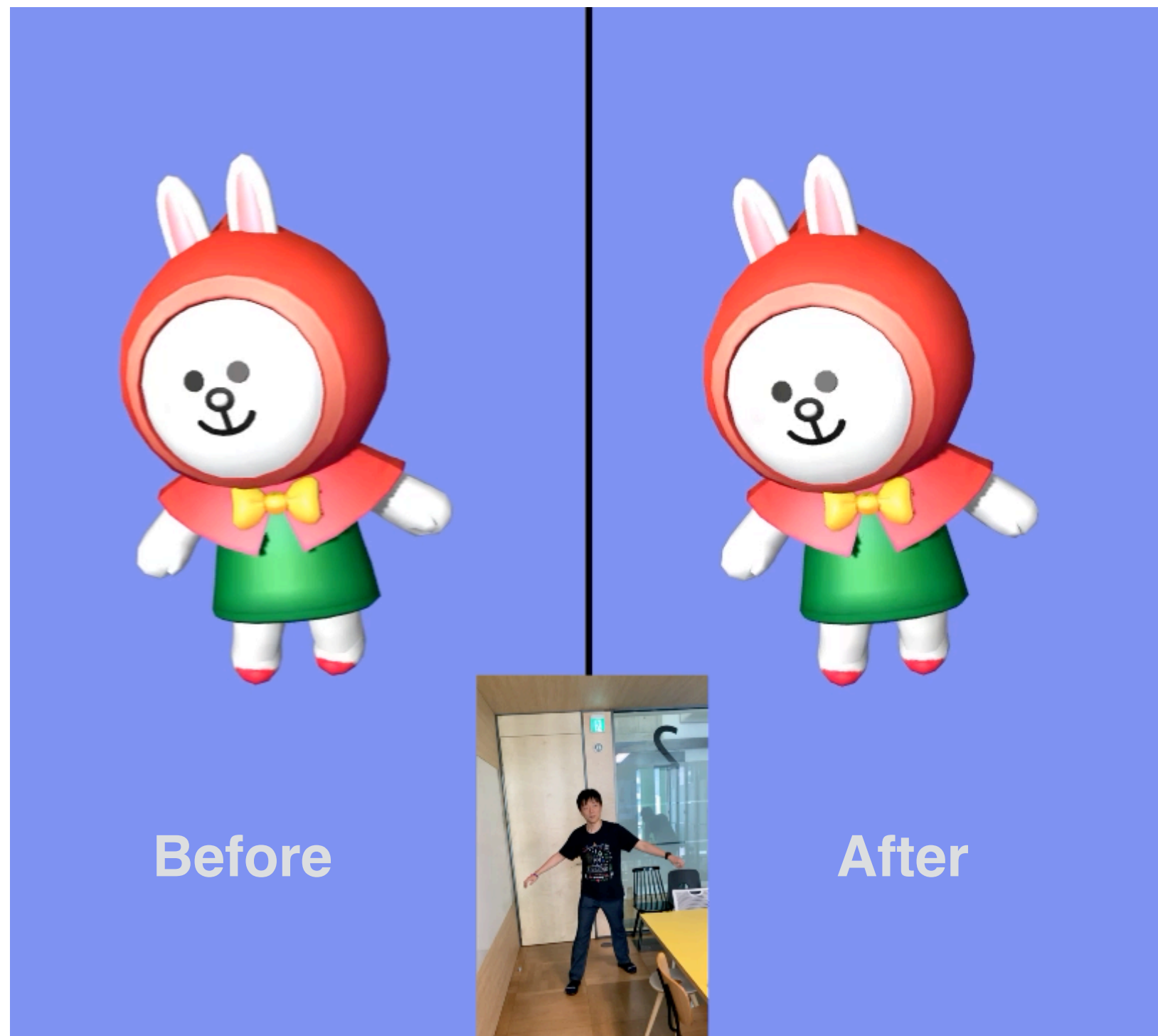
이건 로.보.트.**

삼질 세번째 (March-April, 2019)

VNect가 PC에서 Real-time인데 한번 테스트해볼까?

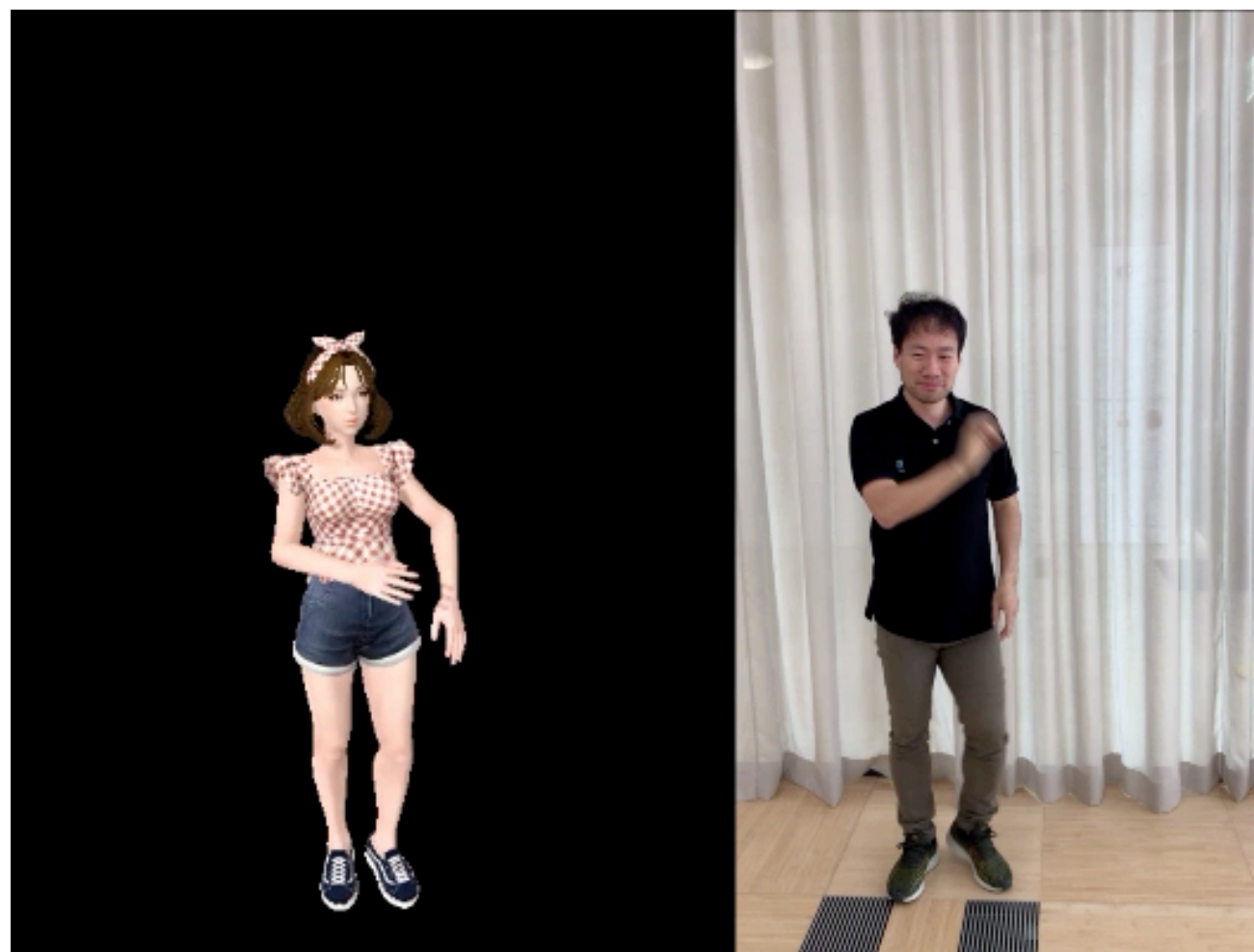
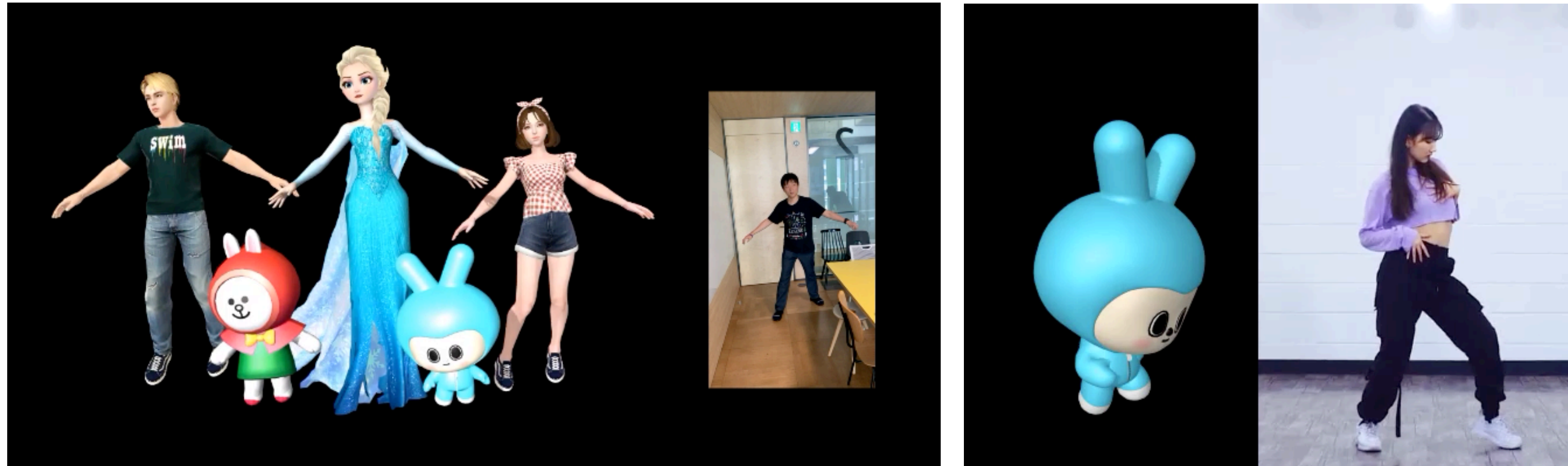


삼질 네번째 (May, 2019)



Jitter를 잡아보자!
이제 아주~ 조금 괜찮네~!

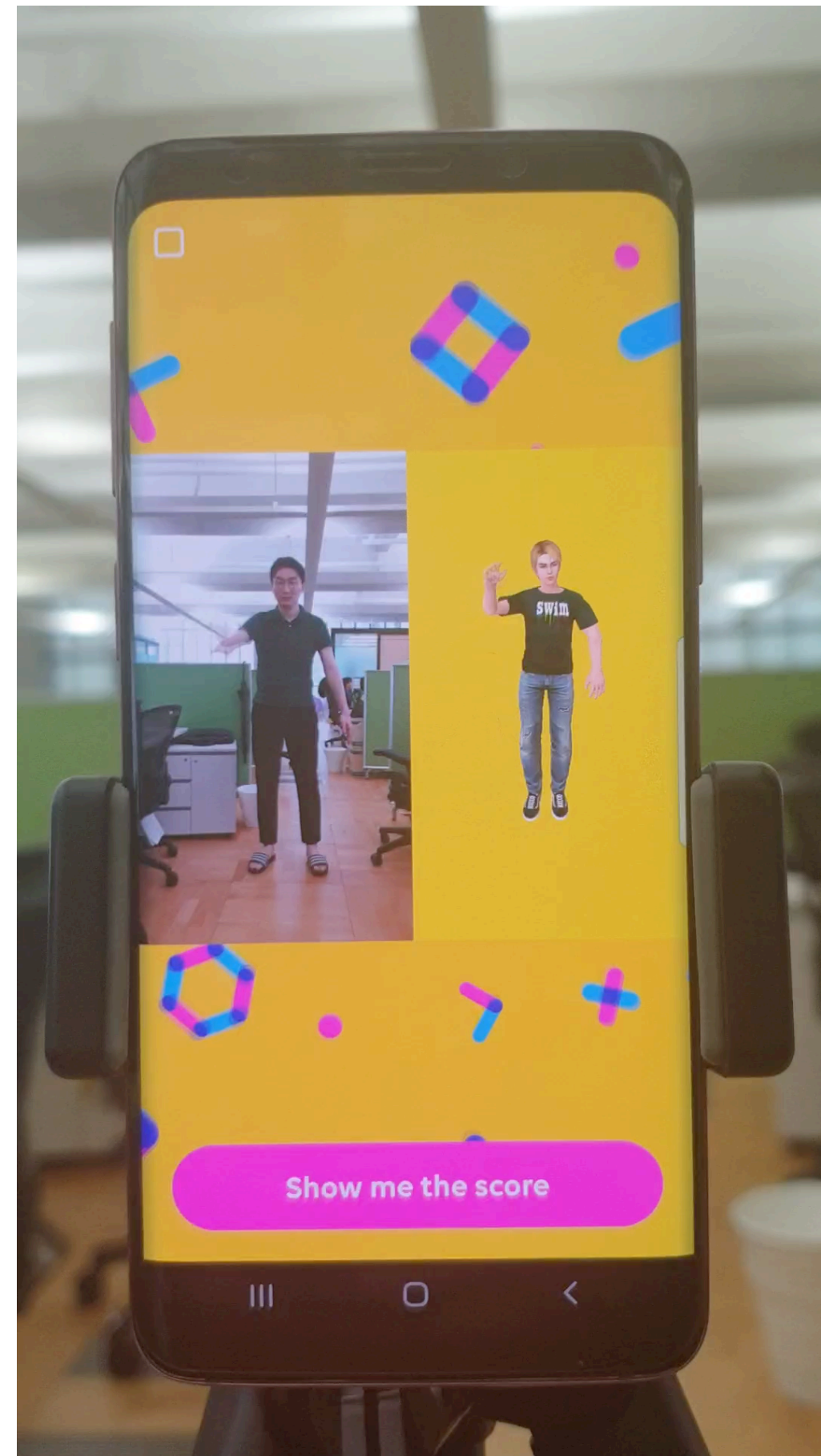
삼질 다섯번째 (June, 2019)



다양한 캐릭터로 테스트 해보자
뭐가 문제인 것일까.. 뭐가 부족한 것일까.

내가 원하는 퀄리티는 무엇일까.
점점 고민에 빠져든다..

삼짇일 여섯번째 (July, 2019)



Server side inference도 해볼까?

모바일에서는 리타게팅만 하고.

→ 서버로 보내고 기다려야 하니깐 되게 지루하다..

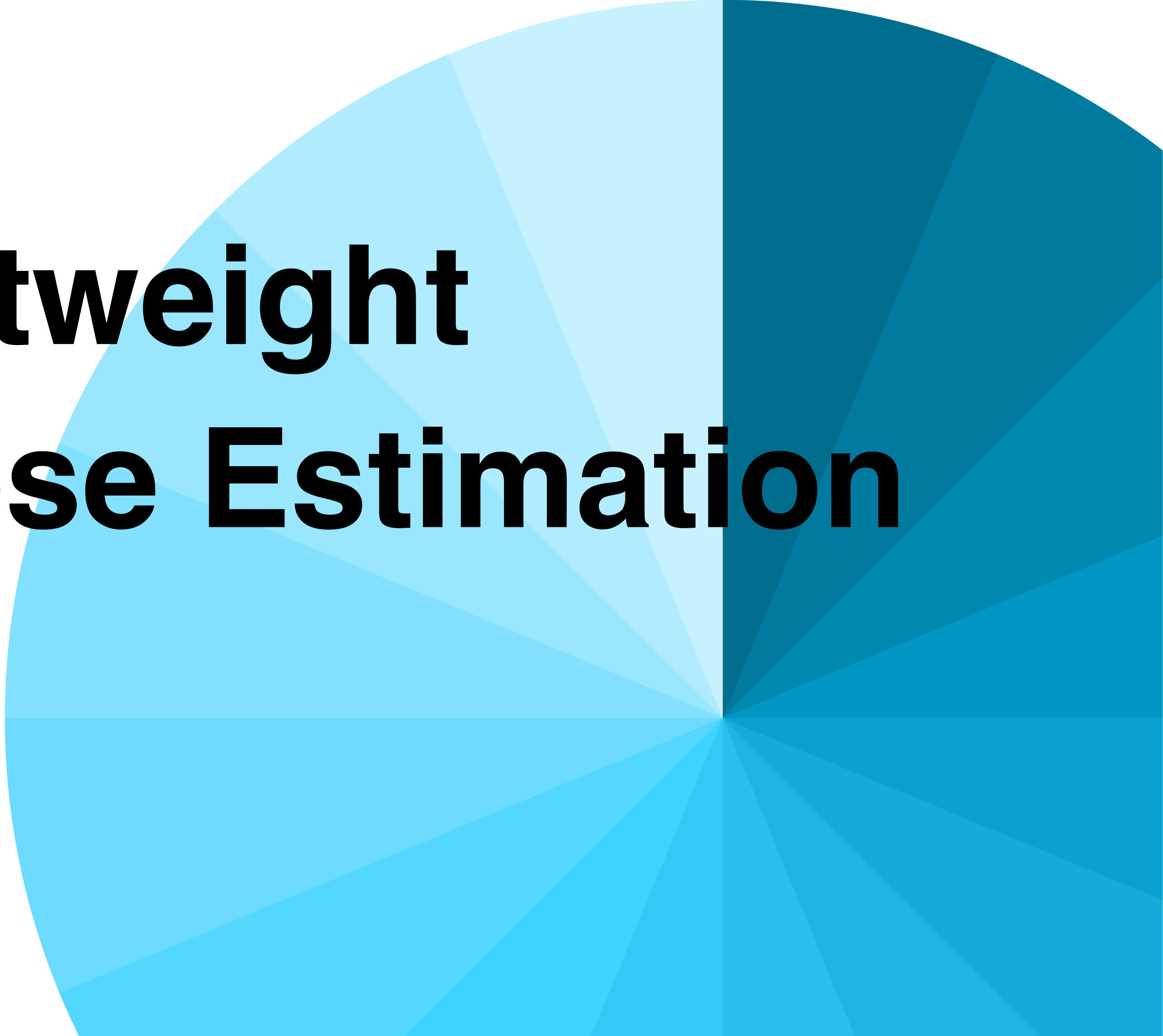
2019년 여름, 자체 모델 개발 착수

삽질, 이제 그만하자!

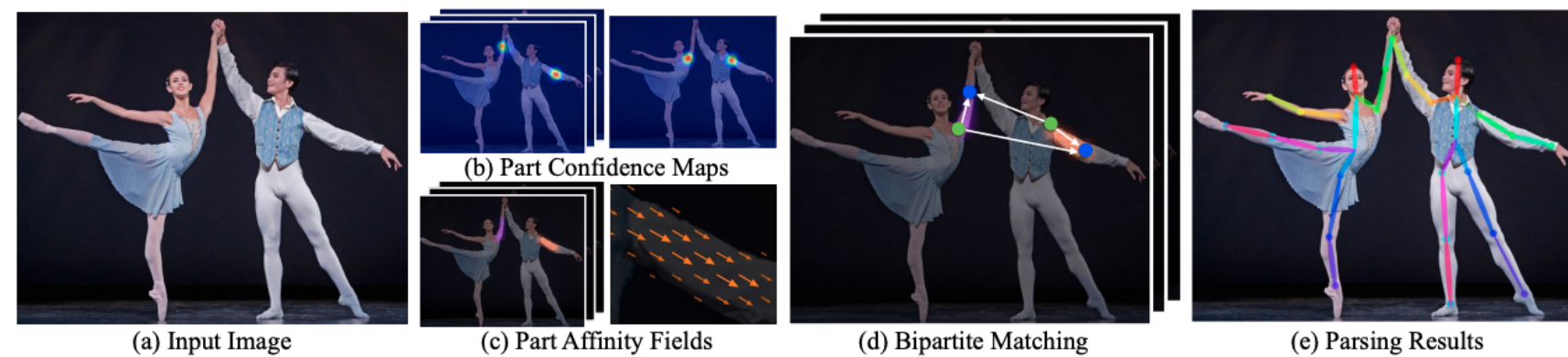
모바일에서 실시간으로 동작하는 모델을 아예 만들자!

SOTA 논문을 보고 직접 테스트 해보면서 땅을 좀 팠더니,
뭐가 문제이고 어디를 개선하면 될 지 알 수 있었다!

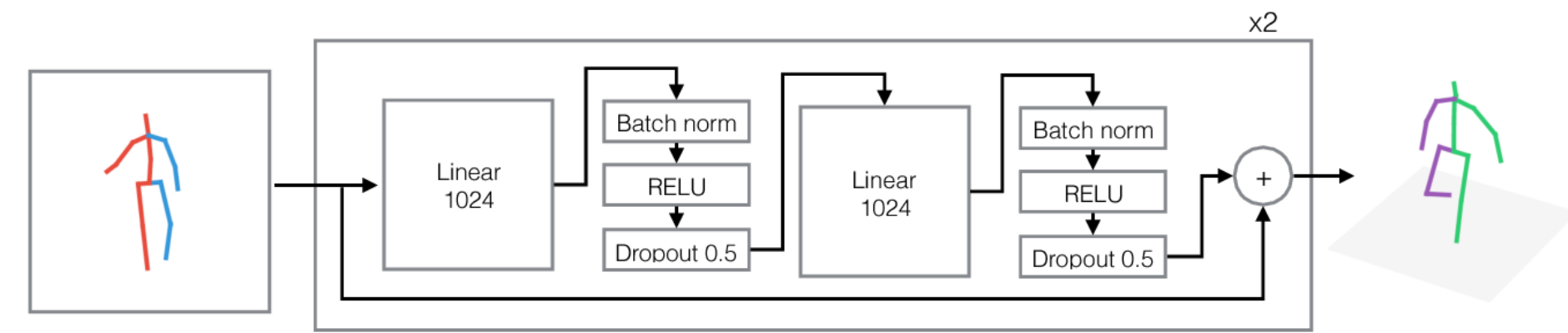
3. Lightweight 3D Human Pose Estimation



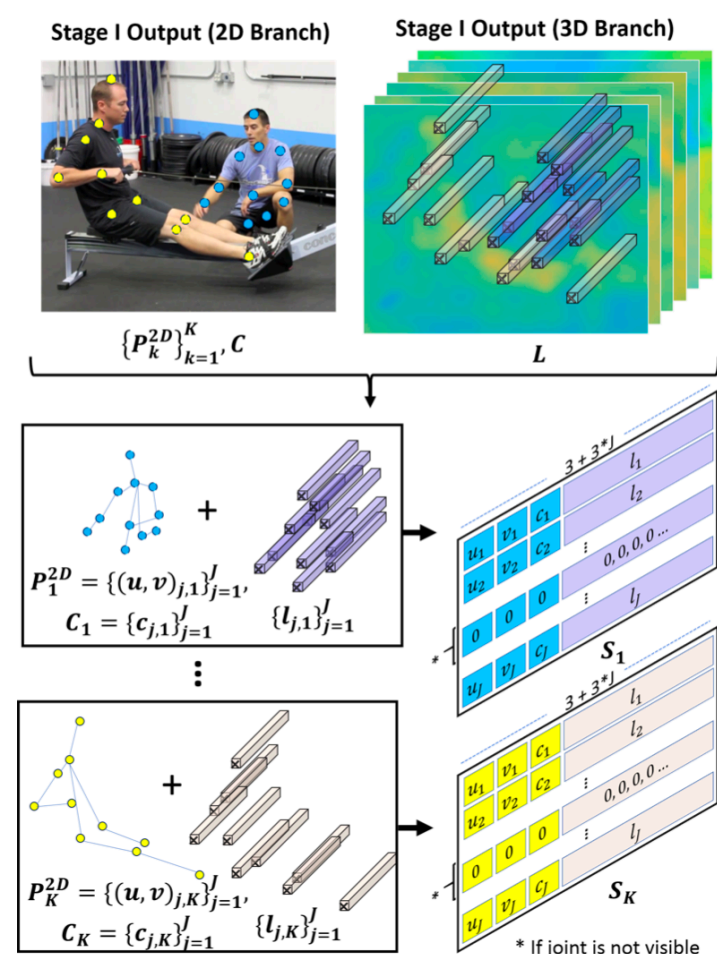
다양한 Human Pose Estimation 모델들..



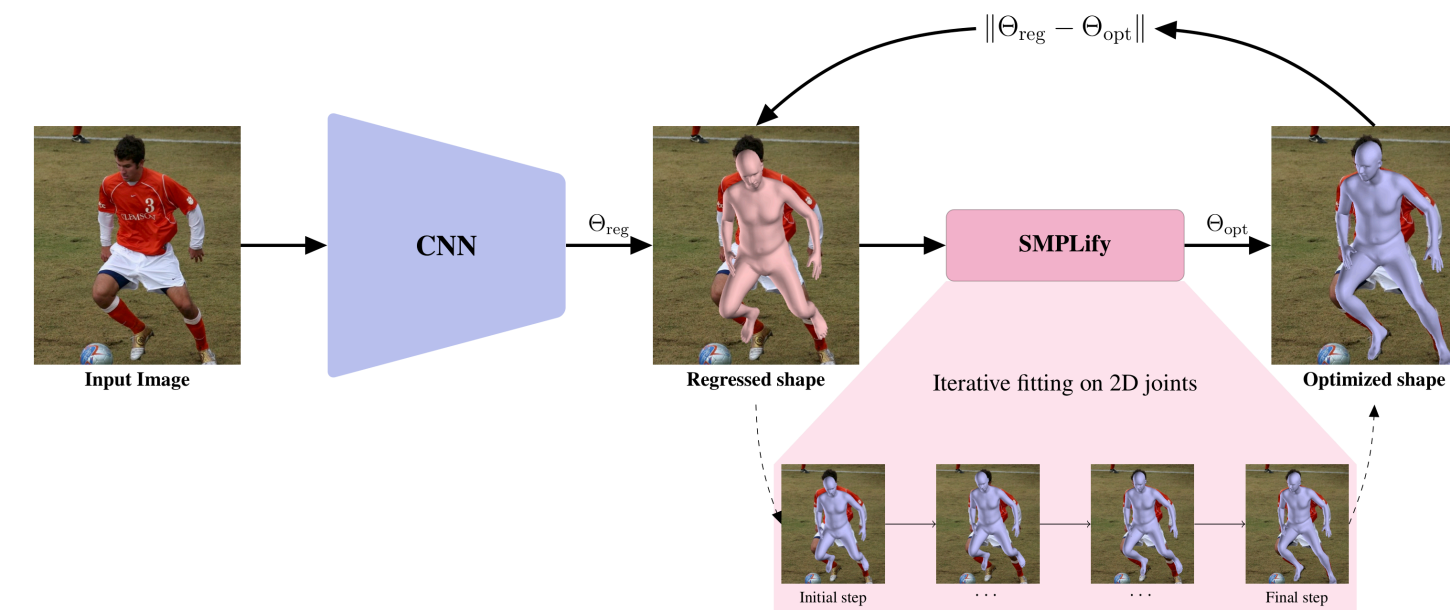
Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields (CVPR 2017)



A simple yet effective baseline for 3d human pose estimation (ICCV 2017)



XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera (SIGGRAPH 2020)



SPIN: Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop (ICCV 2019)

Model 문제 정의, 우리가 추구하는 모델은?

Single Person or Multi-Person

Heatmap (Pose) or SMPL (3D Mesh Reconstruction + Pose)

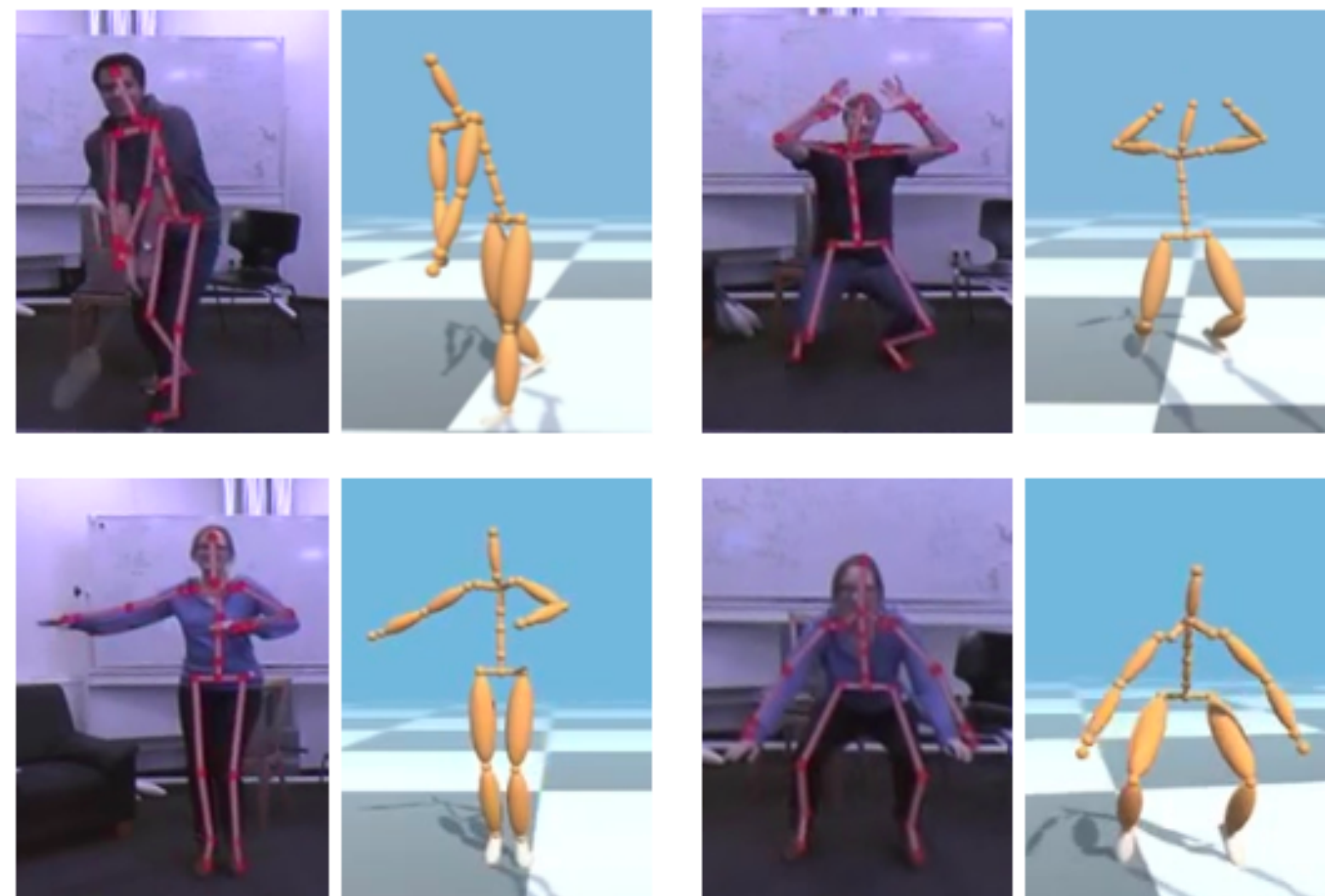
2D Pose Estimation or **3D Pose Estimation**

End to End (Image → 3D) or Lifting (Image → 2D → 3D)

PC (Server) or **Mobile (On-Device)**

3D Human Pose Estimation

VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera (SIGGRAPH 2017)



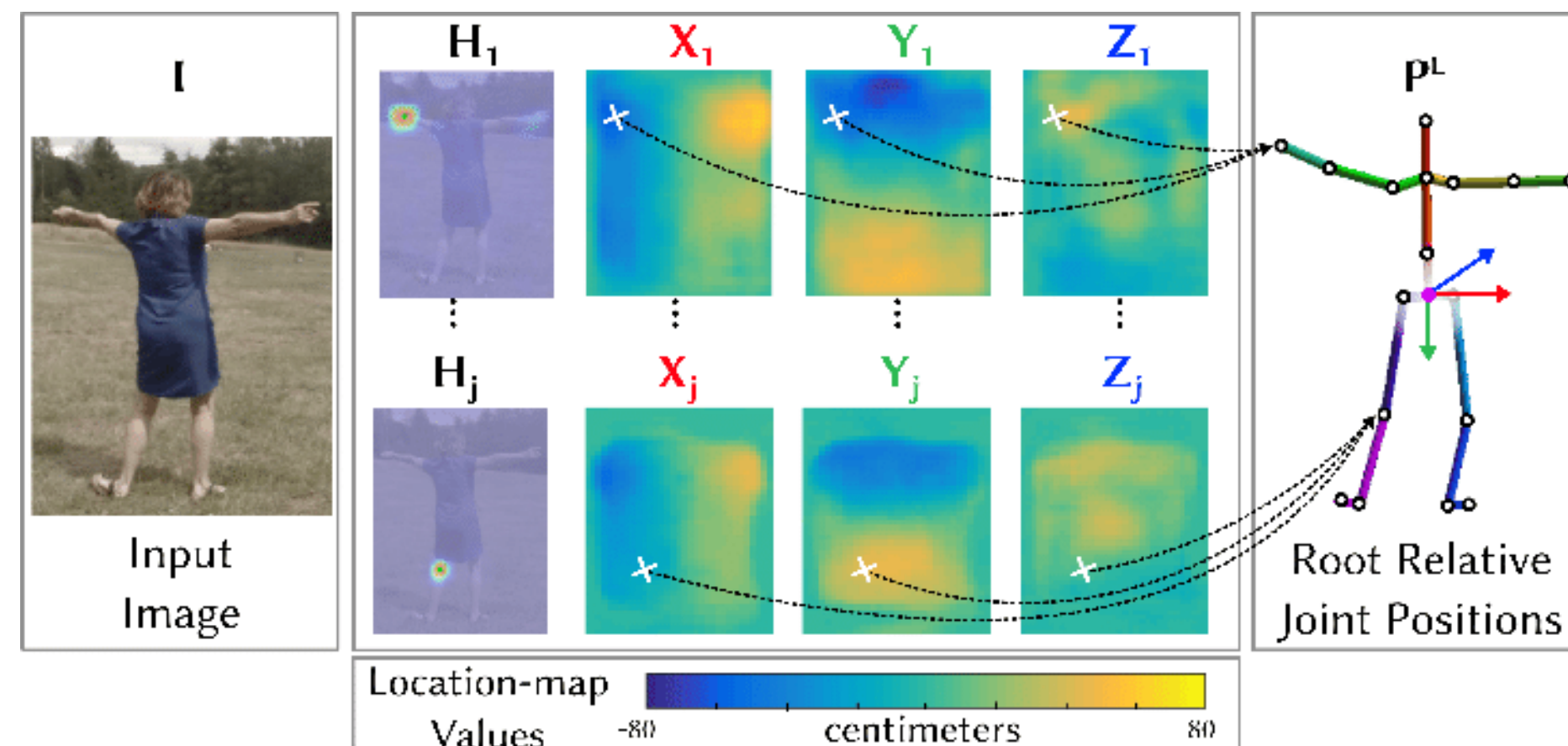
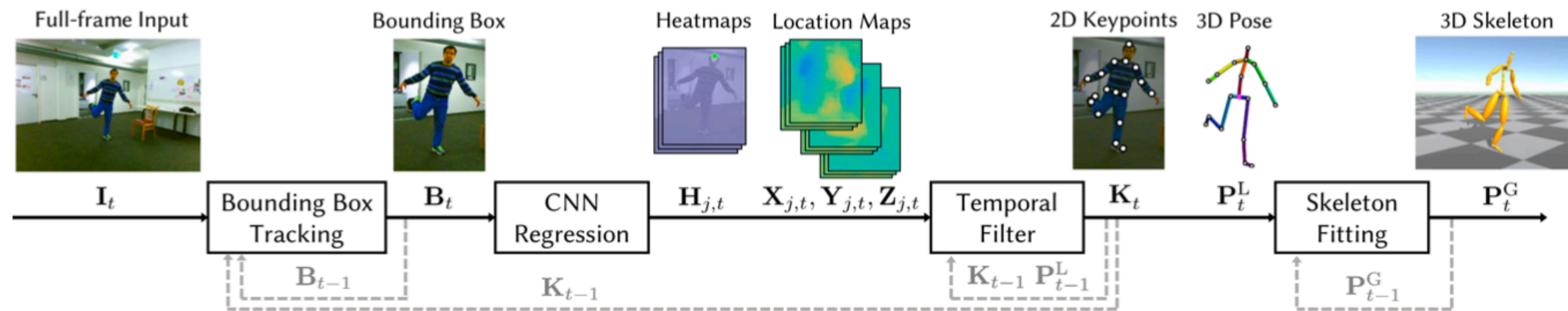
Application to entertainment. The real-time 3D pose estimation method provides a natural motion interface, e.g. for sport games.



Handheld recording with a readily available smartphone camera (left) and our estimated pose (right), **streamed** to and processed by a GPU enabled PC.

3D Human Pose Estimation

VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera (SIGGRAPH 2017)

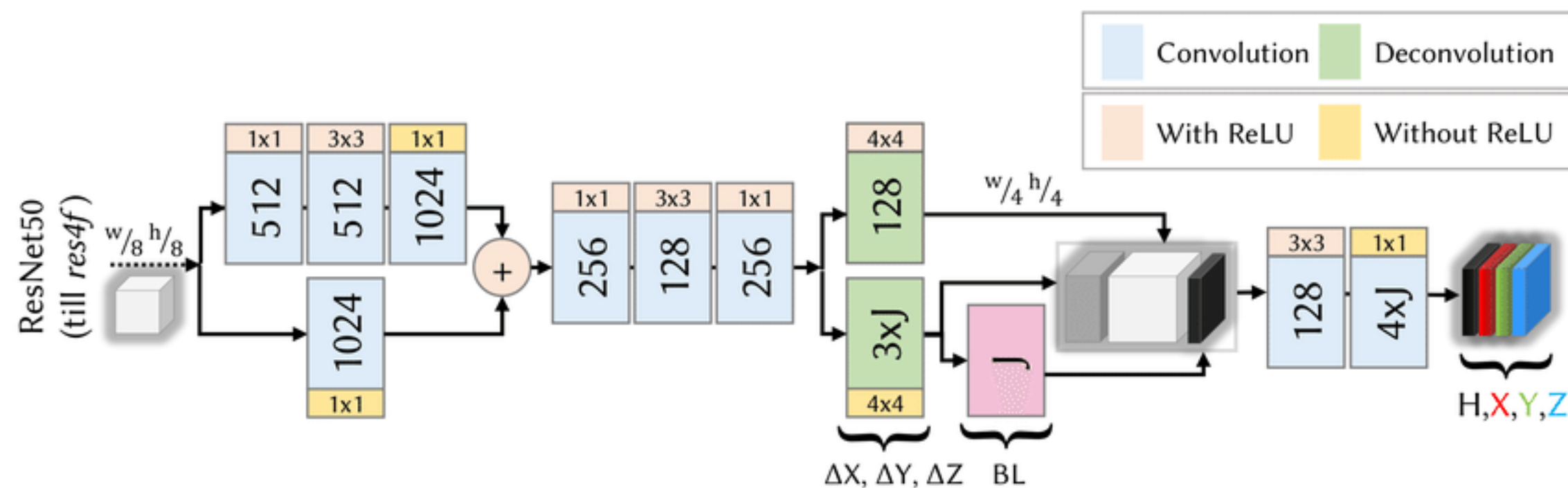


$$\text{Loss}(x_j) = \|H_j^{GT} \odot (X_j - X_j^{GT})\|_2,$$

- 좋은 모델이지만 PC 기반..
- 큰 모델 사이즈 (58.4Mb)
- 5FPS (200ms) on iPhone XS, CPU
- 모바일에서 Realtime으로는 부적합

Lightweight 3D Human Pose Estimation N DEVIEW 2020

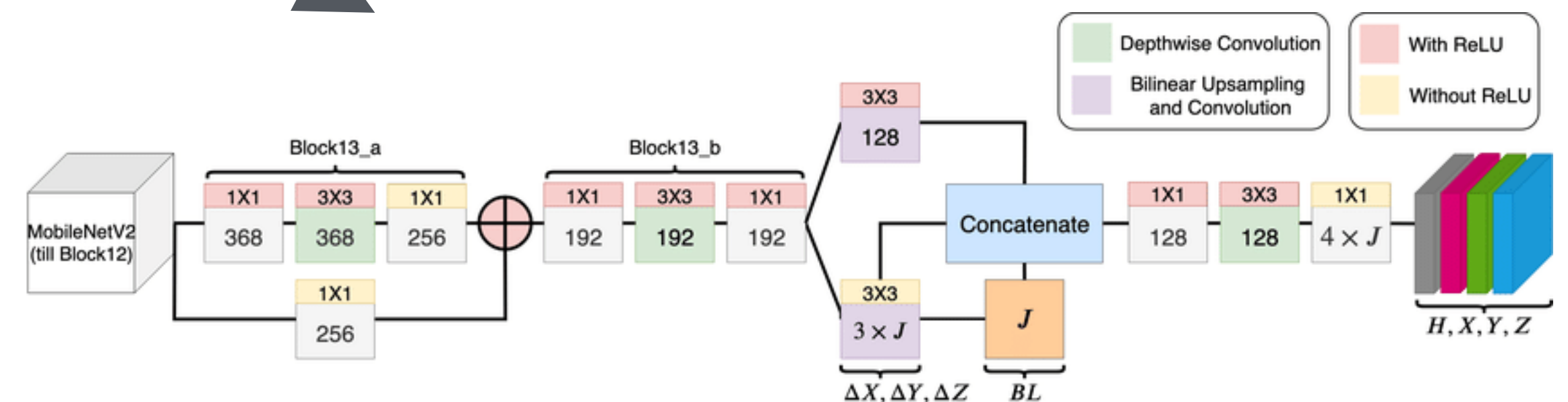
깍자!!



VNect

- ResNet50 → MobileNetV2
- Convolution → Depthwise Convolution
- Deconvolution → Bilinear upsampling and Convolution

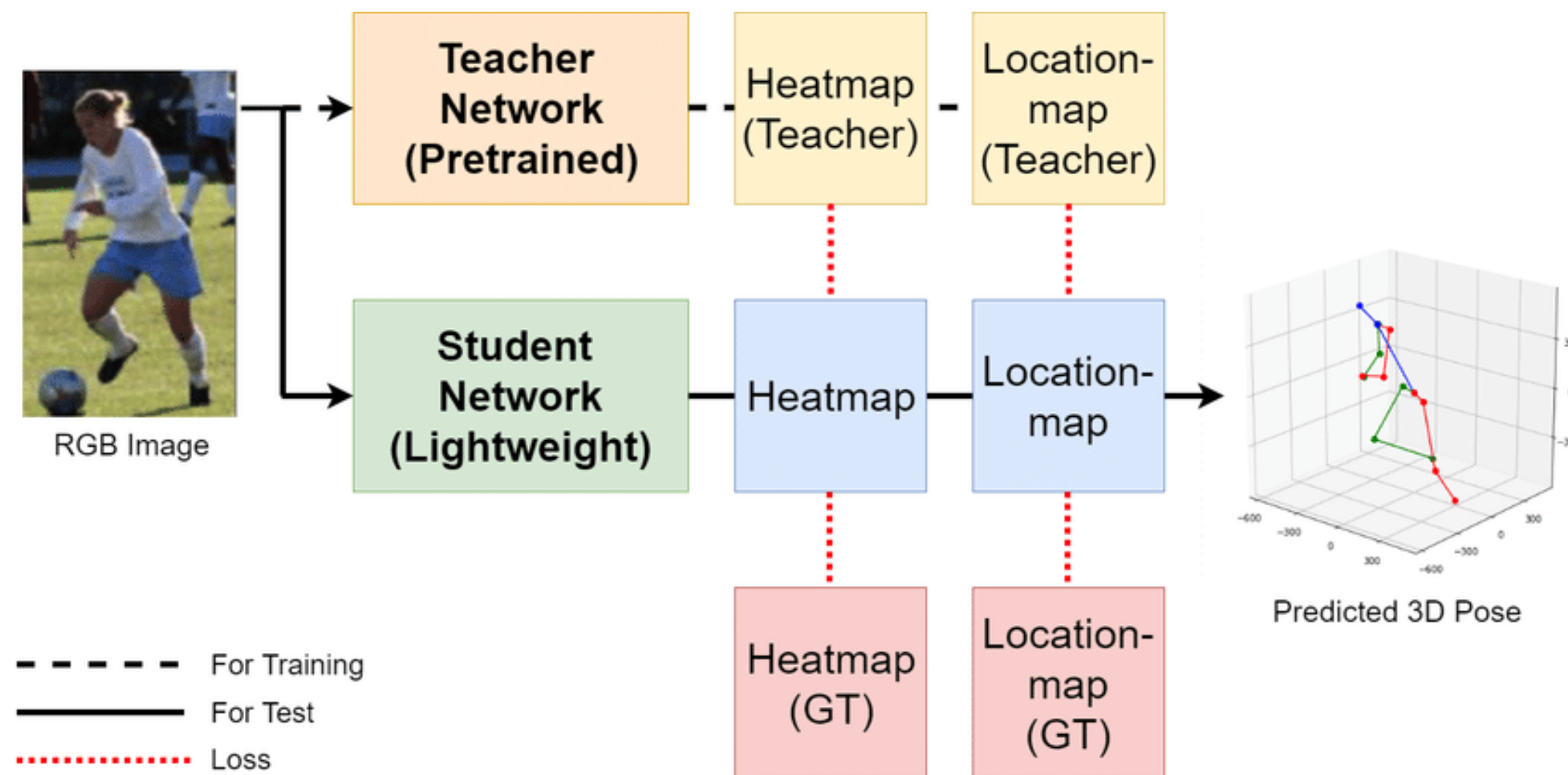
- 14.6M → 1.04M (Parameters), 94% ↓
- 58.4Mb → 4.16Mb (File size)
- 7.3M → 1.36M (FLOPS), 81% ↓



MoVNect

Teacher - Student Learning (Distillation)

모델을 깎으니 Parameter size와 FLOPS는 내려갔지만, 정확도로 같이 내려가는 현상 발생!



$$\mathcal{L}_{HM} = \frac{1}{J} \sum_{j=1}^J \{ \alpha \|H_j - H_j^{GT}\|_2 + (1 - \alpha) \|H_j - H_j^T\|_2 \}$$

$$\mathcal{L}_{LM} = \sum_{j=1}^J \{ \alpha \|H_j^{GT} \odot (L_j - L_j^{GT})\|_2 + (1 - \alpha) \|H_j^{GT} \odot (L_j - L_j^T)\|_2 \}$$

Metric: MPJPE(H36M) *GT와 예측한 조인트 위치간의 거리 평균, 수치가 작을 수록 높은 정확도

80.5mm (VNect) → 113.3mm (MoVNect) ↓

113.3 mm (MoVNect) → 97.3mm (MoVNect Distillation) ↑



Gap: 16.8mm (1.68cm)

Training

- 3D human pose dataset → Indoor images
- Train the network with LSP and MPPII dataset
- Fine-tune the network with Human3.6M and MPI-INF-3DHP dataset
- MPI-INF-3DHP dataset → Background augmentation with Place365 dataset



Number of Joints

Joint	Description
Head Top	머리(정수리)
Neck	목
Right Shoulder	우측 어깨
Right Elbow	우측 팔꿈치
Right Wrist	우측 손목
Left Shoulder	좌측 어깨
Left Elbow	좌측 팔꿈치
Left Wrist	좌측 손목
Right Hip	우측 골반
Right Knee	우측 무릎
Right Ankle	우측 발목
Left Hip	좌측 골반
Left Knee	좌측 무릎
Left Hip	좌측 발목
Pelvis	골반



Comparison of Networks

Methods	Cost-Effectiveness			Inference Time on Devices									
	MPJPE	# Param	FLOPS	iPhone7		iPhone 8		iPhone X			iPhone XS		
				CPU	GPU	CPU	GPU	CPU	GPU	NPU	CPU	GPU	NPU
Mehta et al.[31]	80.5	14.6M	7.3M	275	175	215	140	270	120	120	200	110	17
Martinez et al.[28] w/ SH	62.9	19.3M	22M	750	200	300	160	350	160	160	270	120	20
Kocabas et al.[25]	51.8	34M	14M	500	220	210	210	230	160	160	200	125	50
Ours	97.3	1.03M	1.35M	48	56	40	33	37	28	28	32	22	6

정확도는 1.68cm (80.5mm → 97.3mm) 낮아졌지만, ⬆

파일 사이즈는 94% (58.4Mb → 4.12Mb) 줄어들었으며, ⬇

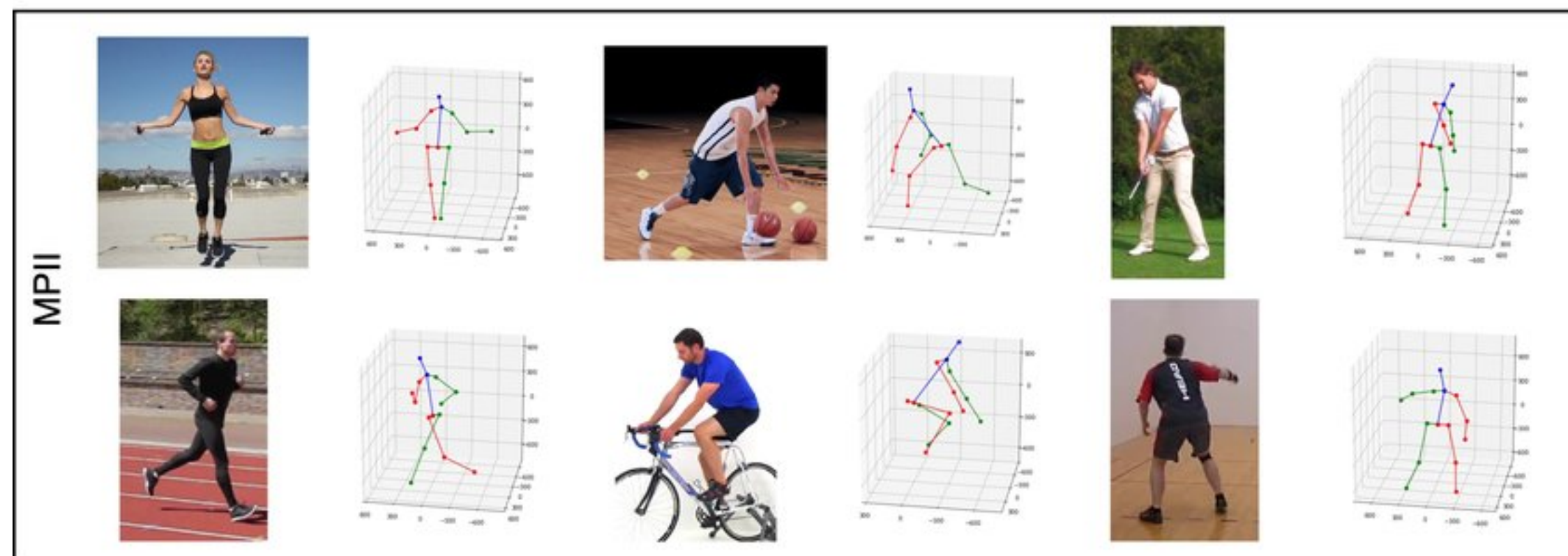
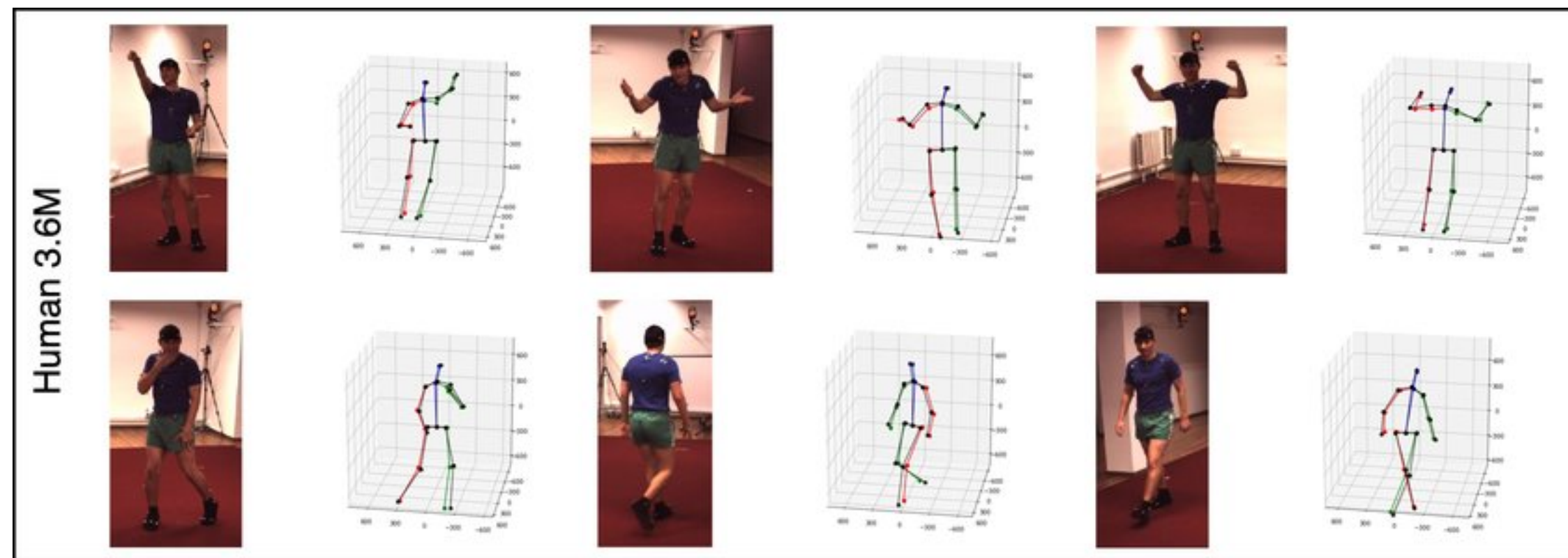
인퍼런스 속도는 iPhone7 (CPU)에서 5.7배 (275ms(3.64FPS) → 48ms(20.83FPS)) 빨라졌고, ⬇

iPhoneXS (CPU)에서는 6.25배 (200ms → 32ms) 빨라졌다! ⬇

모바일에서 실시간 인퍼런스 및 리타게팅이 가능해짐!

정확도도 중요하지만 모바일 환경에서 빠른 인터랙션과 자연스러움도 상당히 중요.

Qualitative Results



Lightweight 3D Human Pose Estimation Network Training Using Teacher-Student Learning

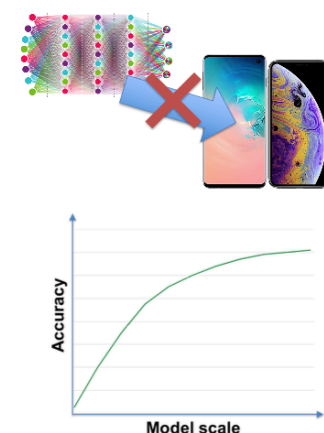


Problem Statement

- The deep neural network (DNN) has achieved remarkable progress in human pose estimation.
 - Without specialized equipment (e.g. MoCap, depth camera, IMU sensor)
 - Robust to the outdoor environment

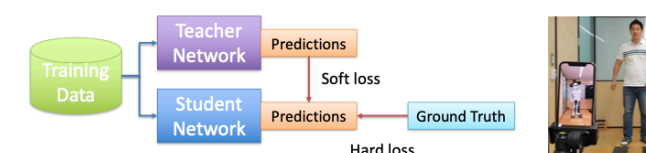


- The performance of DNN highly relies on the model scale.
 - The number of layers, parameters
- Deploying to mobile devices is difficult.
 - The high amount of computation
 - Parallel computing
 - Floating-point operations
- Ways to reduce the number of FLOPS
 - Efficient operation and structure (e.g. Depthwise CNN, linear bottleneck)
 - Reduce the number of parameters
- Reducing amount of parameters loss the accuracy.



Research Goals

- Design a lightweight 3D human pose estimation model.
 - Minimize FLOPS with efficient operation and the small number of parameters.
- Apply the teacher-student Learning method to 3D human pose estimation problem.
 - Increase performance without changing the network structure.
- Deploy to mobile devices to demonstrate the applicability of the proposed method.



Lightweight 3D Human Pose Estimation Network Training Using Teacher-Student Learning

Dong-Hyun Hwang¹, Suntae Kim², Nicolas Monet³, Hideki Koike¹, Soonmin Bae²
Tokyo Institute of Technology¹, NAVER², NAVER LABS Europe³

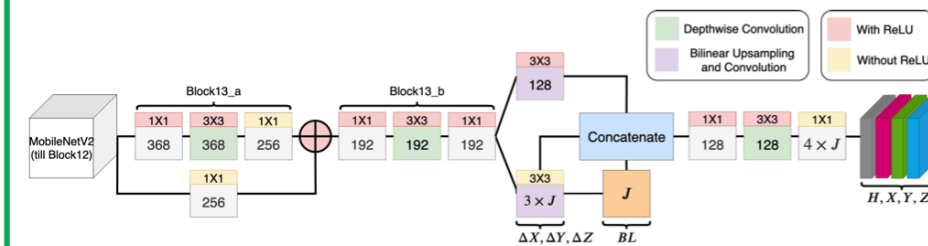


Teaser Video

IEEE 2020 Winter
Conference on
Applications of
Computer Vision



MoVNet Network

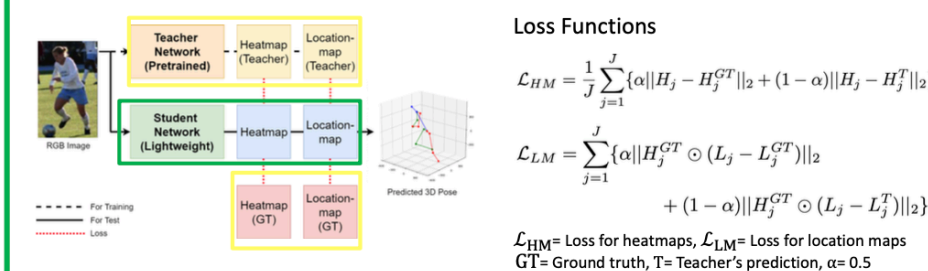


- MobileNetV2 + Depthwise CNNs (1.03M of parameters)

$$BL_j = |\Delta X_j| + |\Delta Y_j| + |\Delta Z_j|$$

- Bone length-features are used as 3D human pose estimation cues.

Teacher-Student Learning based Knowledge Transfer



- Train the student network using the teacher-student learning method.
- Mimicry loss functions transfer the teacher network [Mehta et al. 2017]'s knowledge to the student network.

Post Processing



- 1 Euro filter [Casiez et al. 2012] is used to reduce the temporal jittering.
- Applying kinematics operations transform root-relative joint 3D positions to joint angles with biomechanical plausibility, which can be utilized for 3D character control.

Evaluations

Human3.6M dataset

Methods	Direct	Discus	Eating	Greet	Phone	Photo	Pose	Sitting	SittingD	Sitdown	Wait	WalkD	WalkT	Avg.	# Param	
Zhou et al. 2016	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0
Do et al. 2016	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5	226.9	1120.0	117.7	137.4	99.3	106.5	126.5
Park et al. 2016	100.3	116.2	90.9	116.5	115.3	149.5	117.6	106.9	137.2	190.8	105.8	125.1	131.9	62.6	96.2	117.3
Mehta et al. 2016	52.6	63.8	55.4	62.3	71.8	52.6	72.2	86.2	120.6	66.0	79.8	64.0	48.9	76.8	53.7	68.6
Martinez et al. 2017 w/ SH	51.8	56.2	58.1	59.0	69.5	78.4	55.2	78.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Mehta et al. 2017	62.6	78.1	63.4	72.5	88.3	63.1	74.8	106.6	138.7	78.8	93.8	73.9	55.8	82.0	59.6	80.5
Pavlakos et al. 2018	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Yang et al. 2018	51.5	58.9	59.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Kocabas et al. 2019	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	51.8
Ours	80.6	96.3	92.2	90.4	116.1	82.1	110.9	188.4	224.6	106.9	123.2	98.9	90.4	117.3	80.5	113.3
Ours	72.4	83.4	76.9	82.1	101.9	70.4	91.8	156.5	193.0	92.8	106.4	85.1	76.8	97.2	70.5	97.3 (14%)

- The average MPIPE of our network is 97.3mm (The network has only 1.03M of parameters)

Networks' Cost-effectiveness and inference time

Methods	Cost-Effectiveness			Inference Time on Devices									
	MPJPE	# Param	FLOPS	iPhone7		iPhone 8		iPhone X			iPhone XS		
				CPU	GPU	CPU	GPU	CPU	GPU	NPU	CPU	GPU	NPU
Mehta et al. 2017	80.5	14.6M	7.3M	275	175	215	140	270	120	120	200	110	17
Martinez et al. 2017 w/ SH	62.9	19.3M	22M	750	200	300	160	350	160	160	270	120	20
Kocabas et al. 2019	51.8	3.4M	14M	500	220	210	210	230	160	160	200	125	50
Ours	97.3	1.03M	1.35M	48	56	40	33	37	28	28	32	22	6

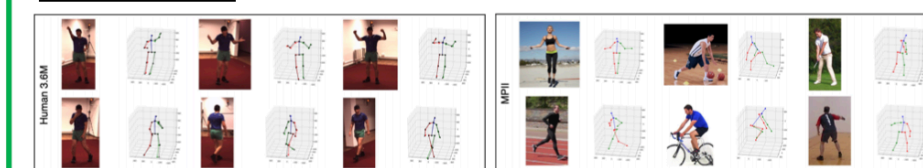
- At least 19 up to 160 FPS of inference performance on mobile devices.
- GPU can be fully utilized for graphics rendering because of real-time performance on CPU.

Effectiveness of the teacher-student learning method

Teacher-Student Learning	MPIPE
N	113.3
Y	97.3 (14%)

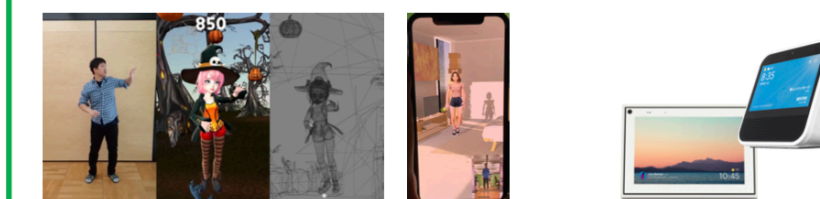
- 14% accuracy improvement is observed with the proposed method.

Qualitative Results



Applications

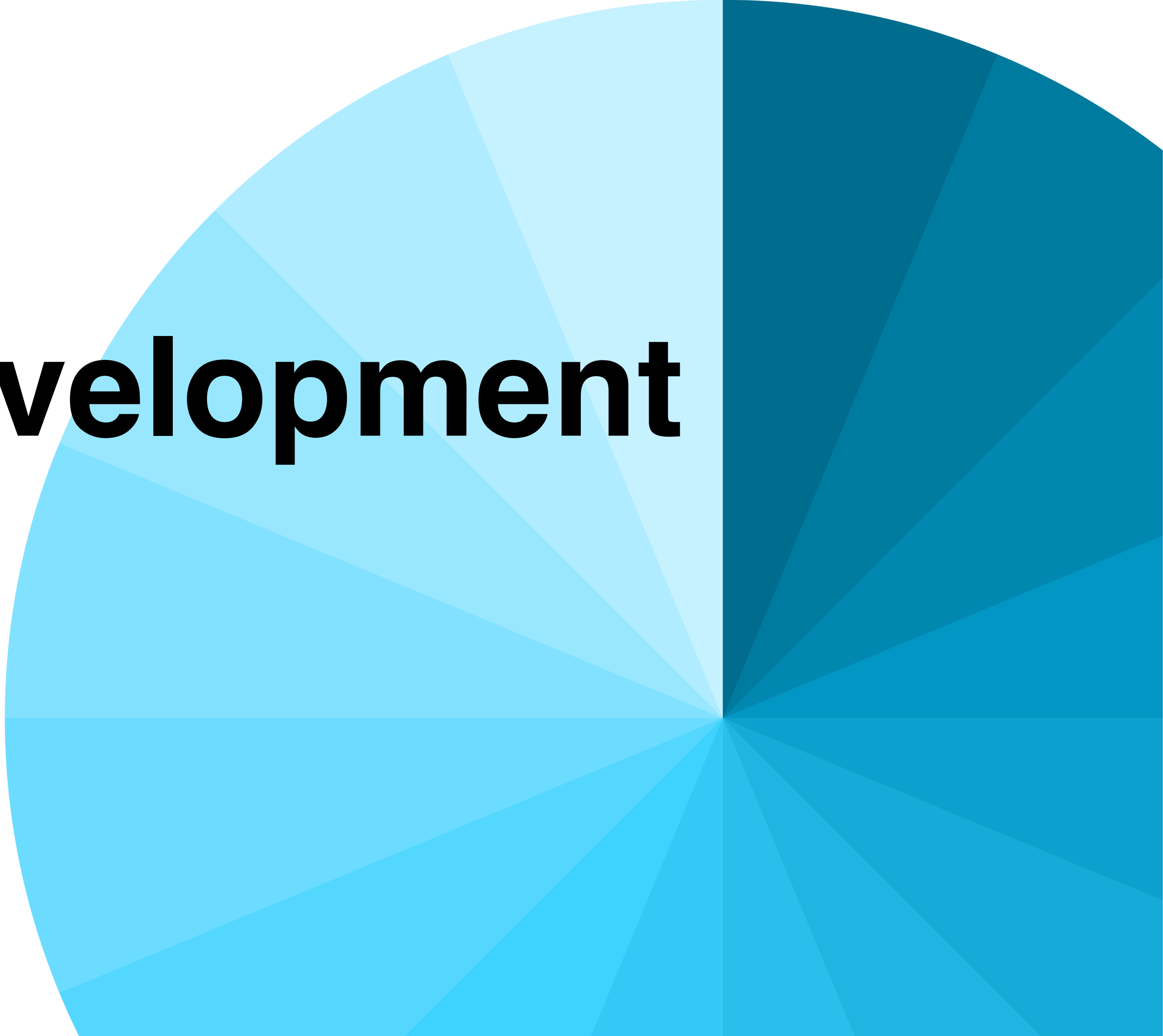
- Interactive AR/VR applications
- Simply Accessible Motion Capture



<https://arxiv.org/abs/2001.05097>

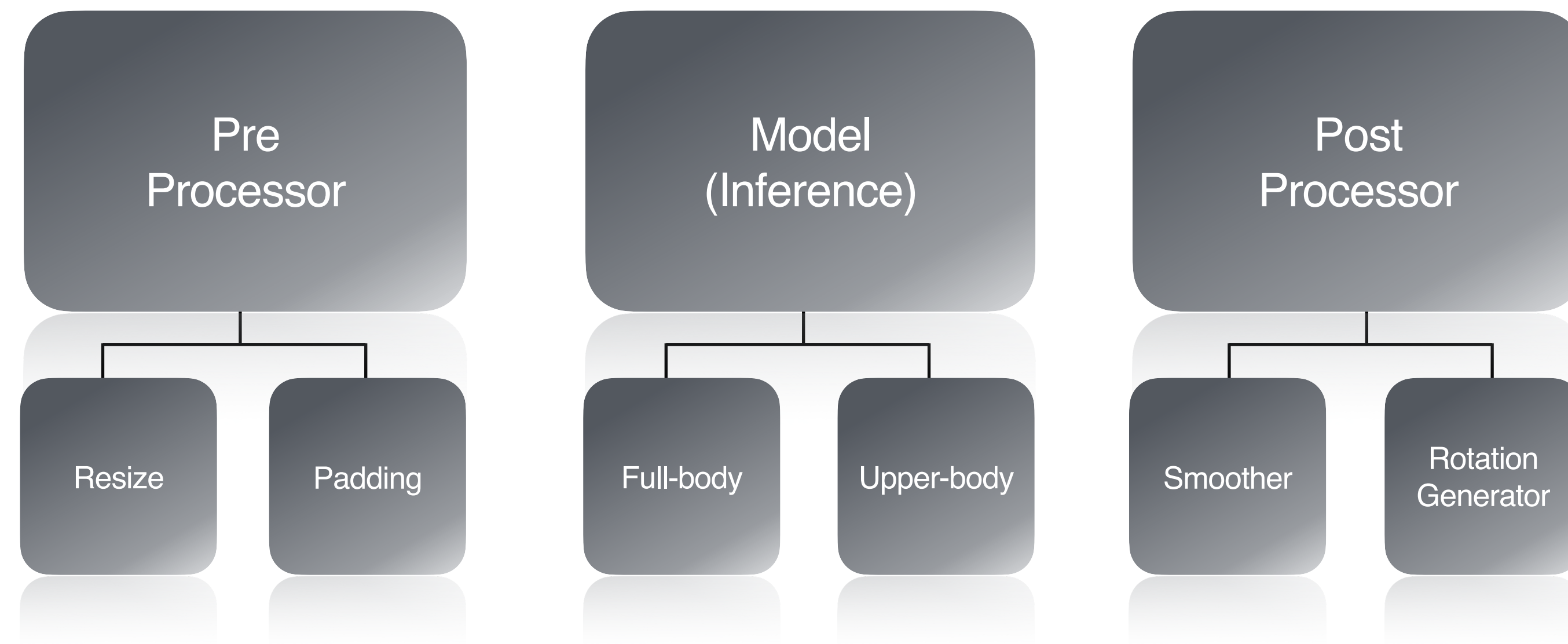
IEEE Winter Conference on Applications of Computer Vision (WACV)

4. SDK Development



Component Based Pipeline

Attach and detach components

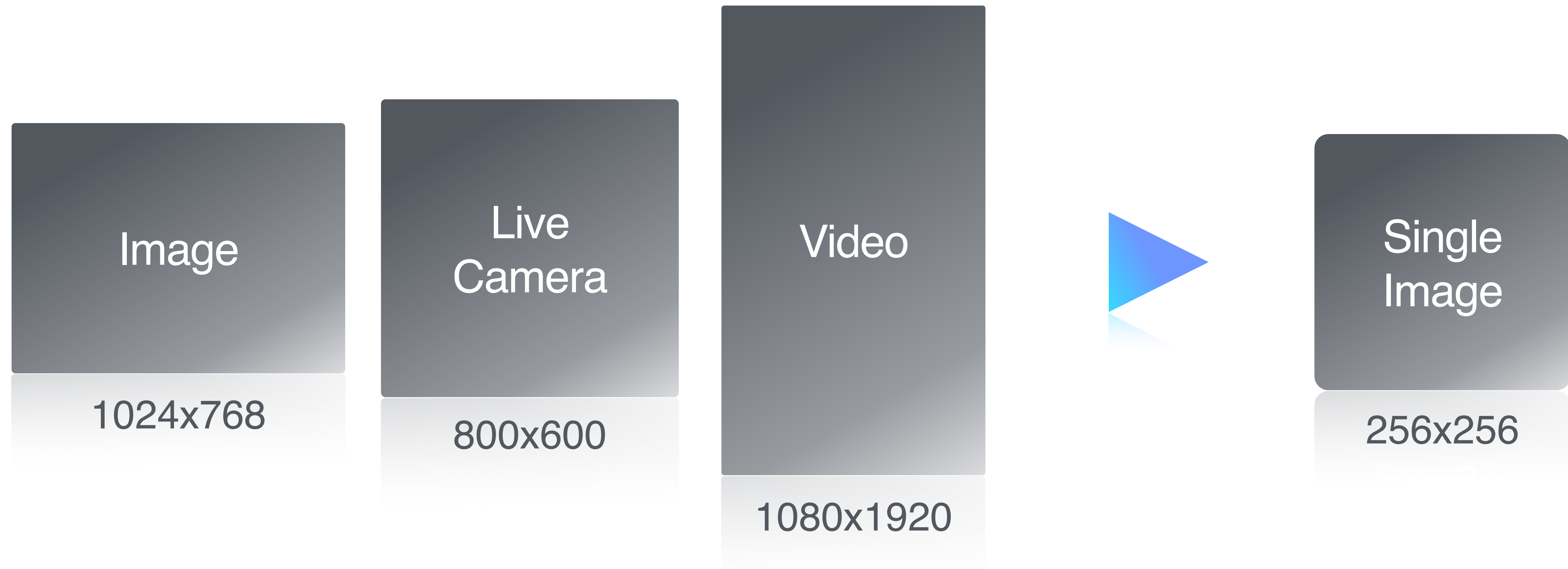


```
class PipelineCoreML : public Pipeline {
public:
    PipelineCoreML(const PipelineInferenceFuncType& infer_func, const int authTool = 0) :
        Pipeline("LSNet (coreML)")
    {
        preprocess.attach(new PipelineComponentObjcPrepare());
        preprocess.attach(new PipelineComponentResize());
        preprocess.attach(new PipelineComponentPadding());

        inference.attach(new PipelineComponentInferCoreML(infer_func));

        postprocess.attach(new PipelineComponentSmooth());
        postprocess.attach(new PipelineComponentMakeQuaternions(authTool));
    }
};
```


Change the image size to feed into Neural Network



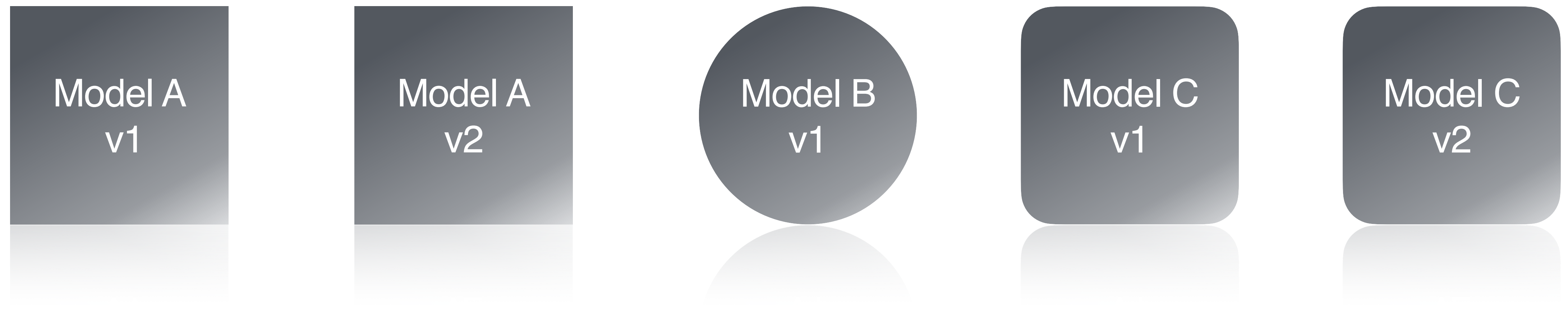
Encryption and Decryption

To protect neural network information



Downloadable Model

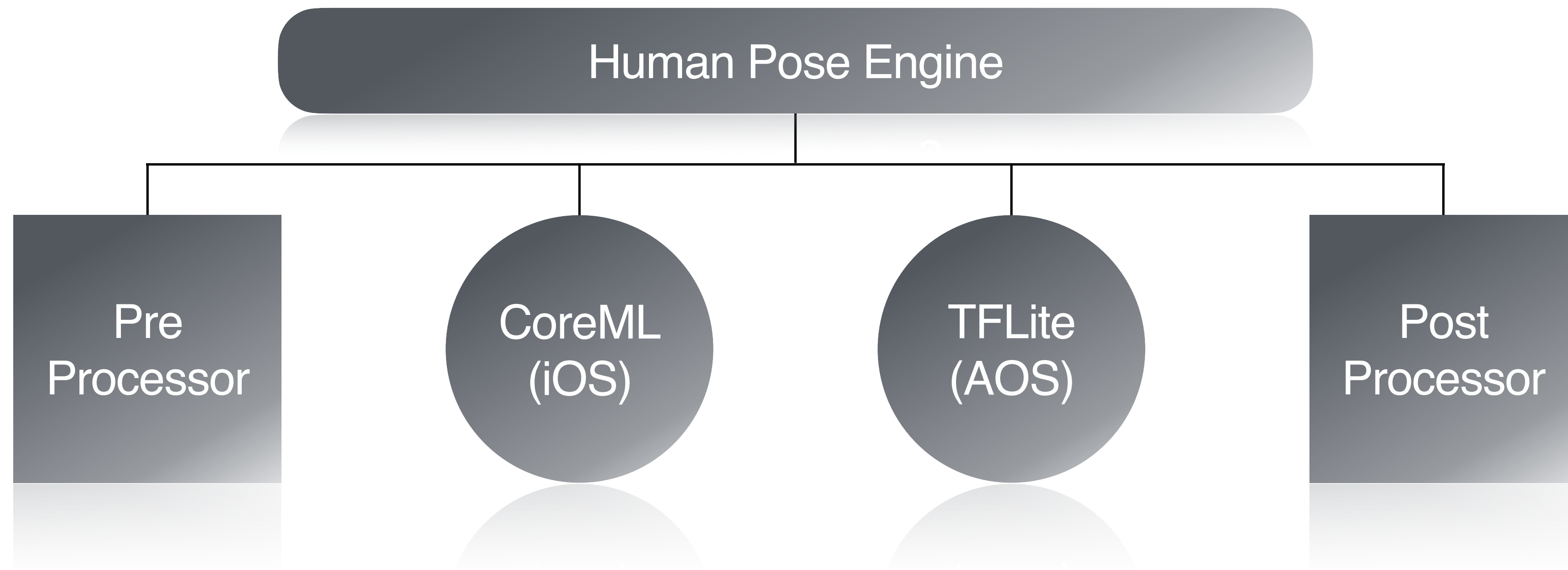
- **Various version of Full-Body and Upper-Body models**
- **Efficient for updating and deploying**



Save at the local internal directory

Inference Framework

- **Support iOS and Android devices**
- **Common pre/post processor in C++**



Post Processor

- **Interpolation, Extrapolation and Reducing the jitter**
- **Attach and Detach filters**

1Euro

Kalman

Animation
Blending

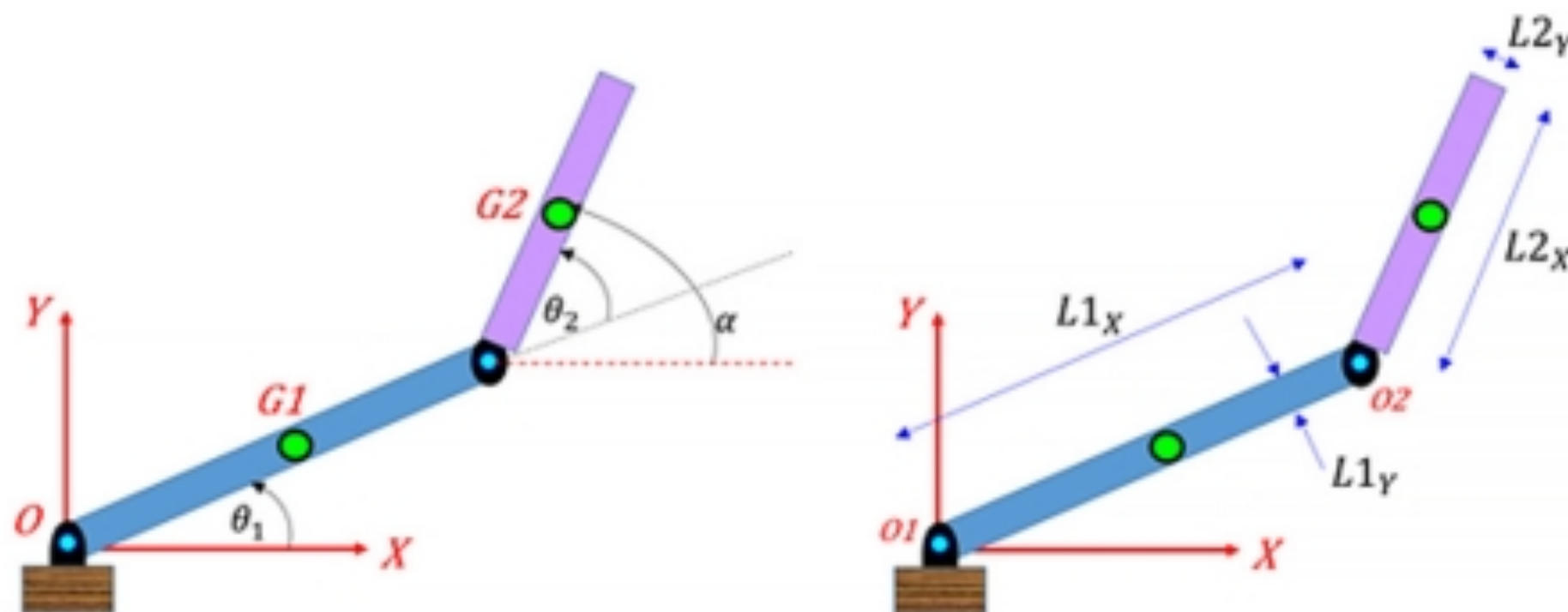
Abnormal
Pose

IK
(Inverse
Kinematics)

Rotation Generator

모델로 부터 예측된 결과는 3D 좌표값(위치값)인데 어떻게 회전 값을 구하지?

Analytical Inverse Kinematics

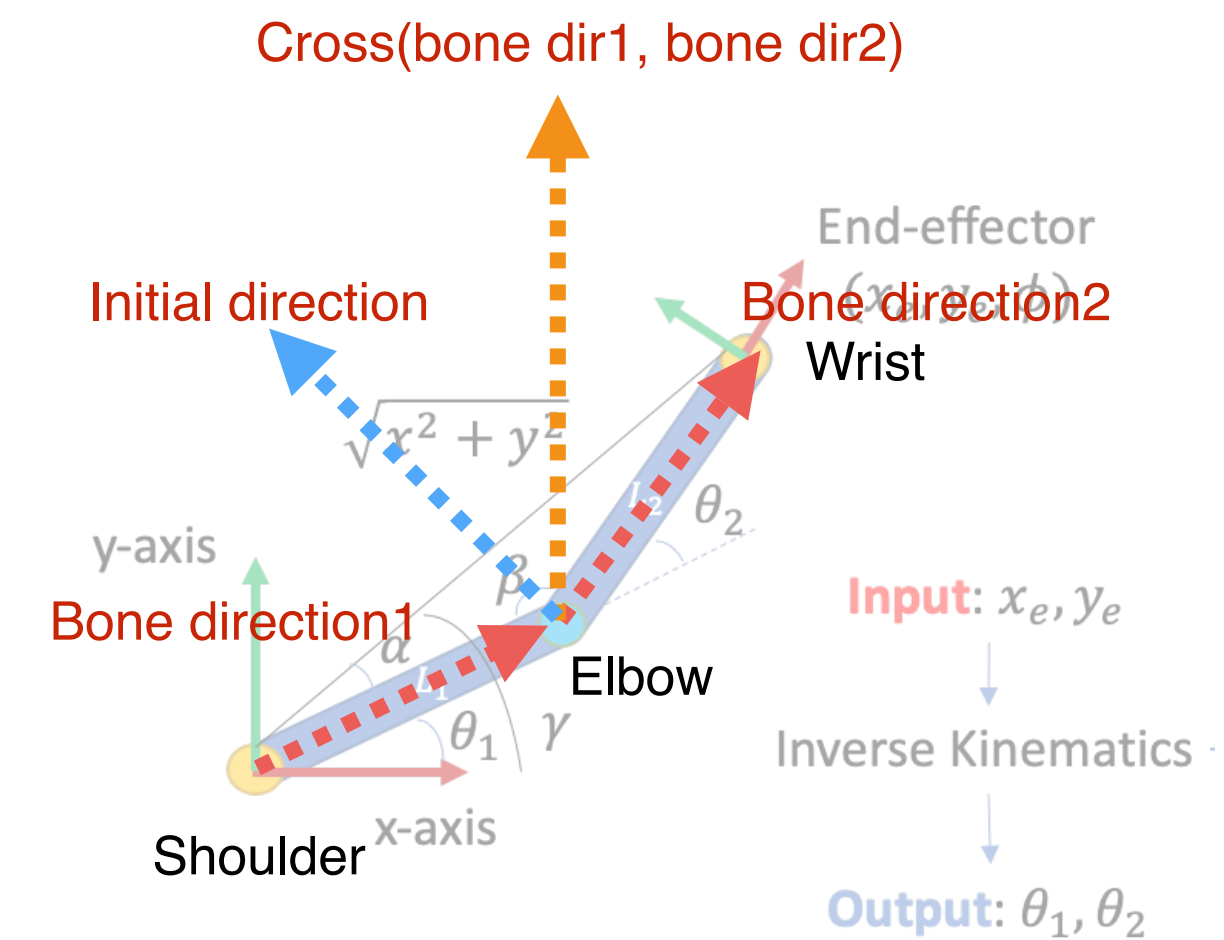
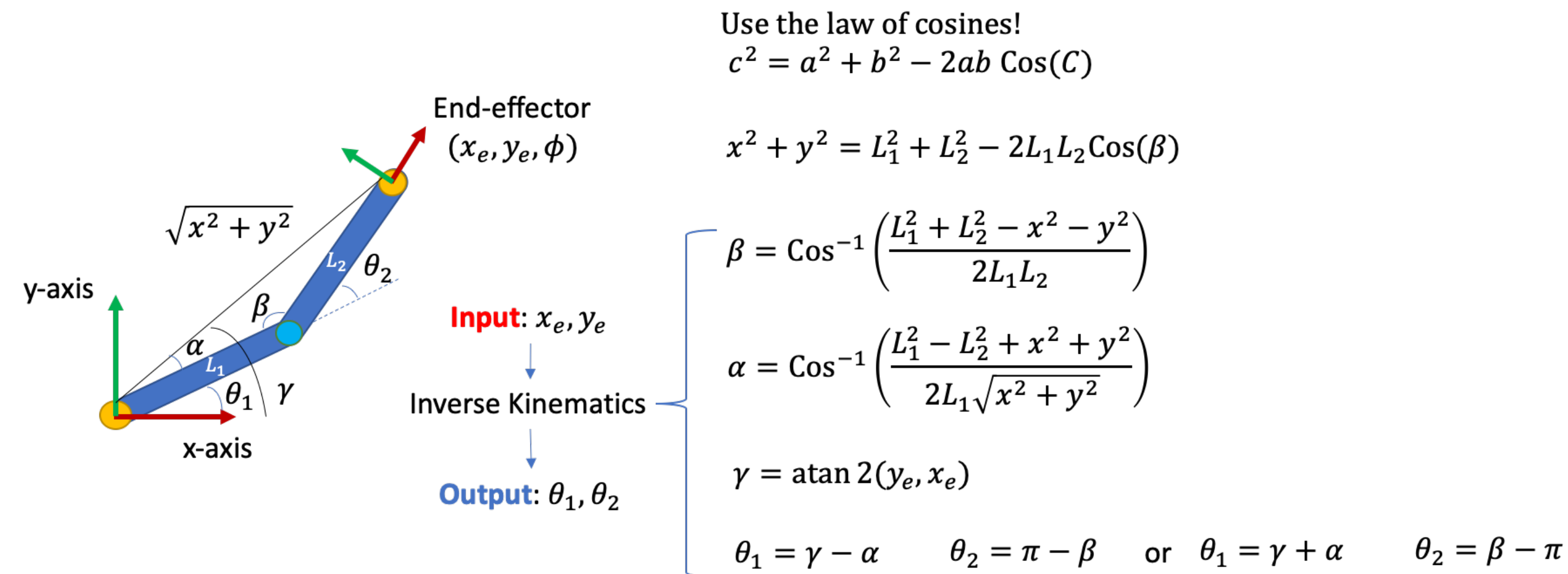


Each joint angle is calculated from the pose of the end-effector based on a mathematical formula. By defining the joint parameters and end-effector poses symbolically, IK can find all possible solutions of the joint angles in an analytic form as a function of the lengths of the linkages, its starting posture, and the rotation constraints.

Analytical IK is mainly used for robots with low degrees of freedom (DoF) due to the nonlinearity of the kinematics equations and the lack of scalability for redundant robot configurations.

Rotation Generator

- Define the joint space
- Calculate without Rotation constraints

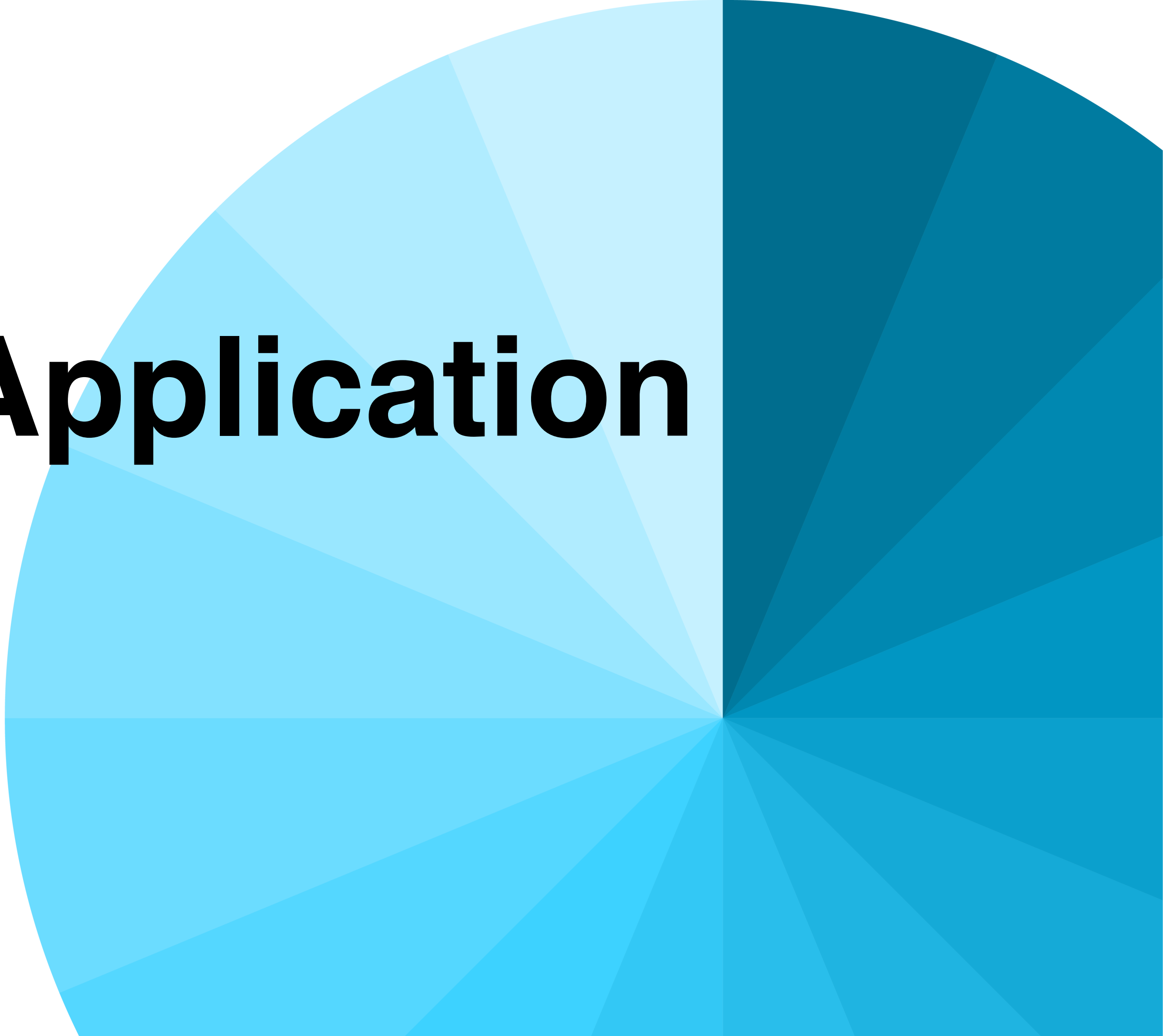


There are one more solutions and a scale issue

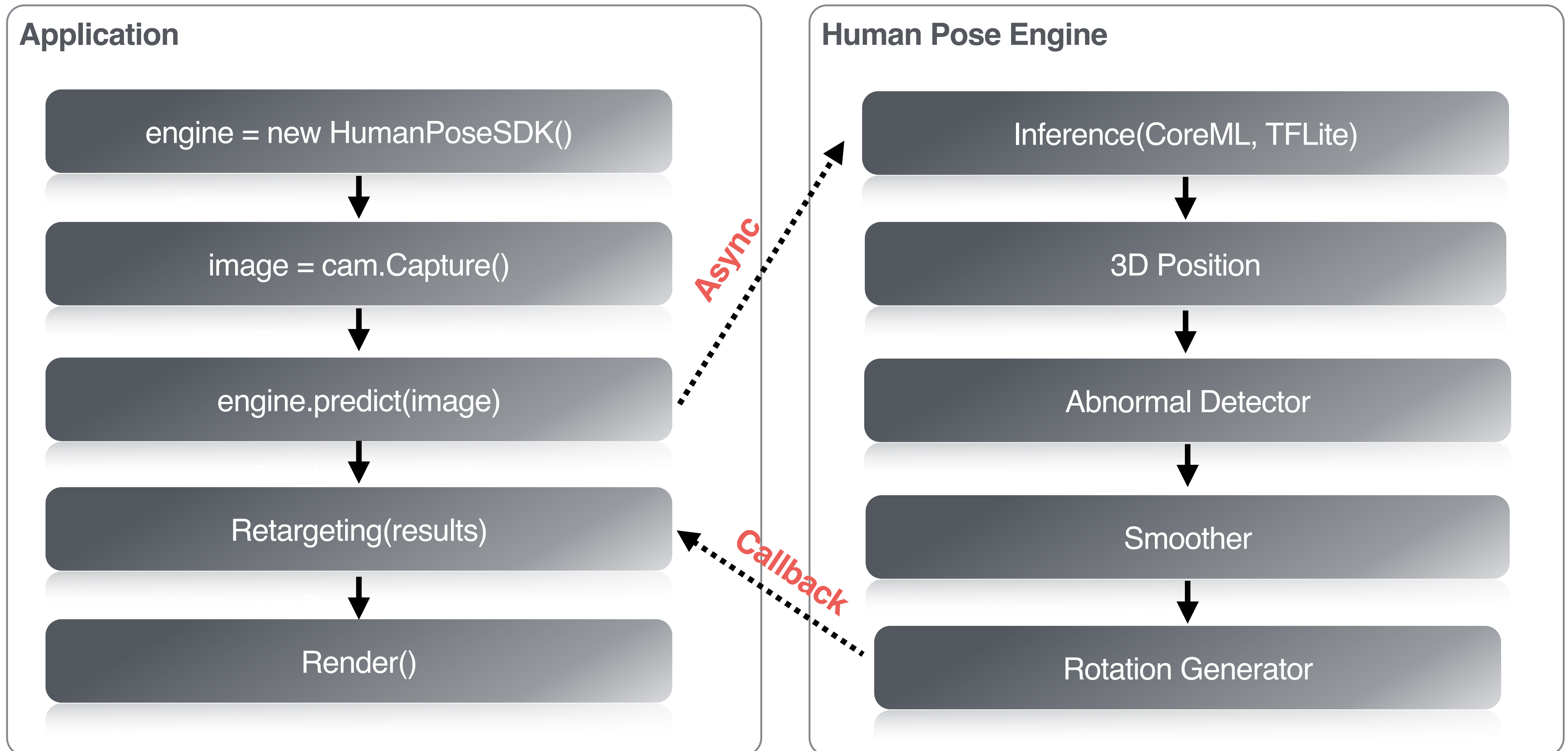
Output Format

Properties	Type	Description
numberOfPeople	Int	검출된 사람 수
threshold	Float	모델별 신뢰 임계값
jointConfidences	Vector<float>	관절별 신뢰값
isAbnormalPose	Bool	정상, 비정상 자세 유무
jointPositions2D	Vector<Vector2>	2차원 좌표값
jointQuaternions3D	Vector<Vector3>	3차원 좌표값
jointQuaternions	Vector<Quaternion>	회전값
performanceMeasure	Struct	성능 측정값

5. Mobile Application



SDK 연동하기



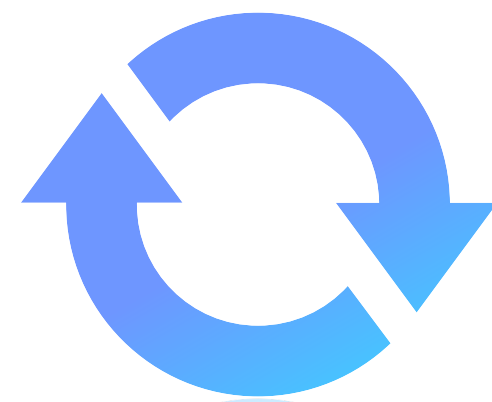
SDK API 사용 및 Application 이벤트 핸들링

N DEVVIEW
2020

비동기 SDK API와 Callback 함수 그리고 Retargeting

Asynchronous API

```
private func request(image: UIImage?) {  
    guard let image = image else { return }  
    if let engine = humanPoseCoreML {  
        let humanPoseRequest = HumanPoseEngineRequest()  
        humanPoseRequest.image = image  
        engine.process(humanPoseRequest)  
    }  
}
```



Render

Callback

```
func onComplete(didPredictComplete result: HumanPoseEngineEstimationResult) {  
    avatar.retarget(result: result)  
  
    self.skeleton2dView.drawSkeleton(result: result)  
    self.skeleton2dView.drawSkeleton(result: result)  
    self.skeleton3dView.update(positions: result.positions3D)  
}
```



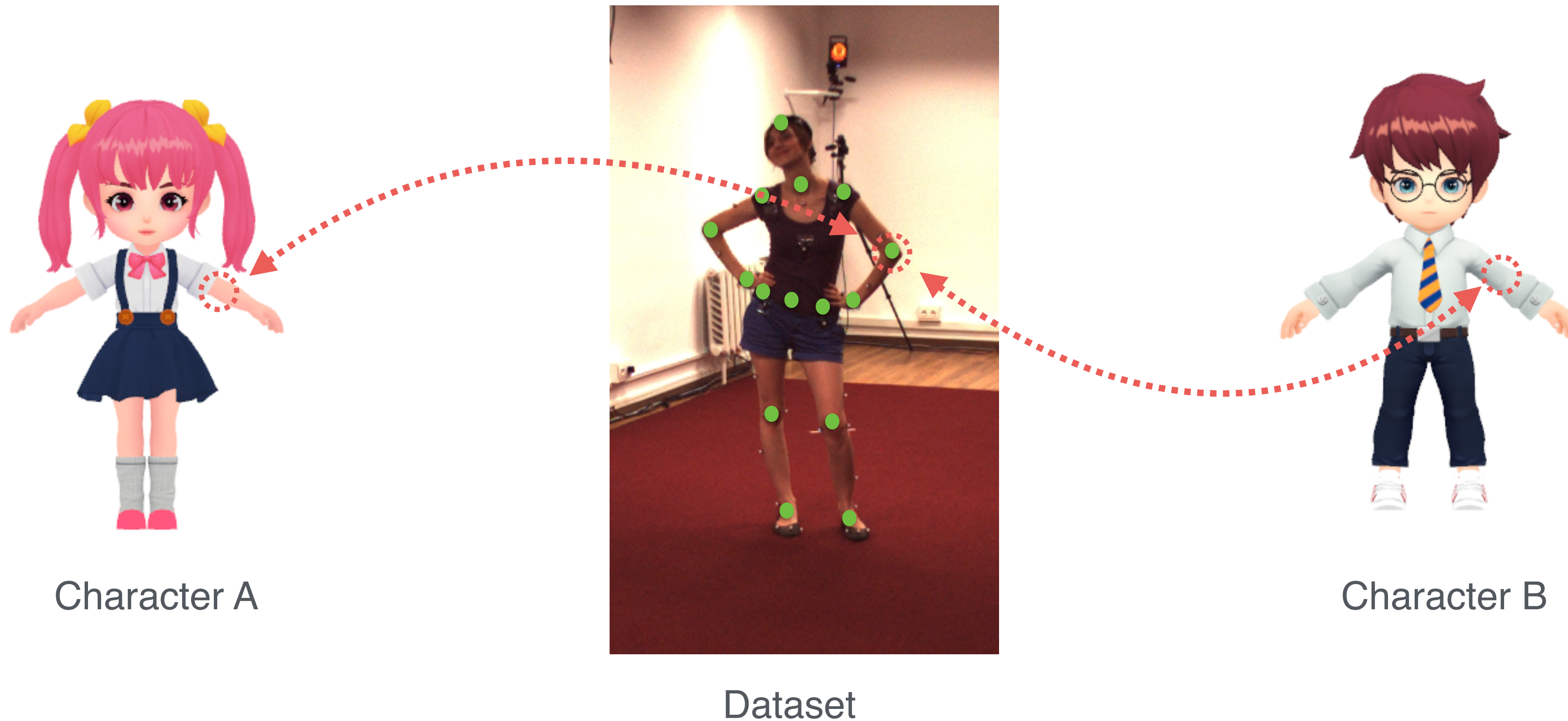
Retarget

```
func retarget(result: HumanPoseEngineEstimationResult) {  
    let quaternions = result.rotations  
    setRotation(type: HumanBodyBones.Neck, quaternion: quaternions[kHPEJointNeck])  
    setRotation(type: HumanBodyBones.RightUpperArm, quaternion: quaternions[kHPEJointRightShoulder])  
    setRotation(type: HumanBodyBones.RightLowerArm, quaternion: quaternions[kHPEJointRightElbow])  
    setRotation(type: HumanBodyBones.LeftUpperArm, quaternion: quaternions[kHPEJointLeftShoulder])  
    setRotation(type: HumanBodyBones.LeftLowerArm, quaternion: quaternions[kHPEJointLeftElbow])  
    setRotation(type: HumanBodyBones.Spine, quaternion: quaternions[kHPEJointSpine])  
}
```



3D 캐릭터 애니메이션 리타게팅

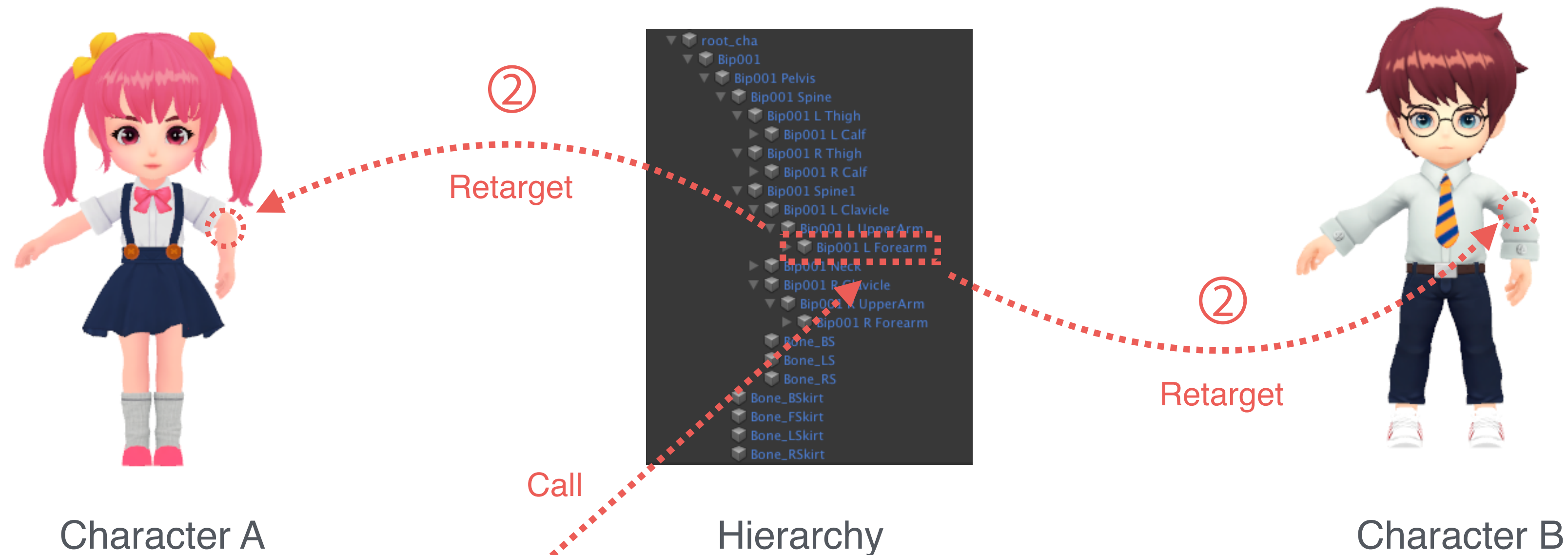
Character skeleton structure and Mapping



The 3D character must have the corresponding joints to the dataset

3D 캐릭터 애니메이션 리타게팅

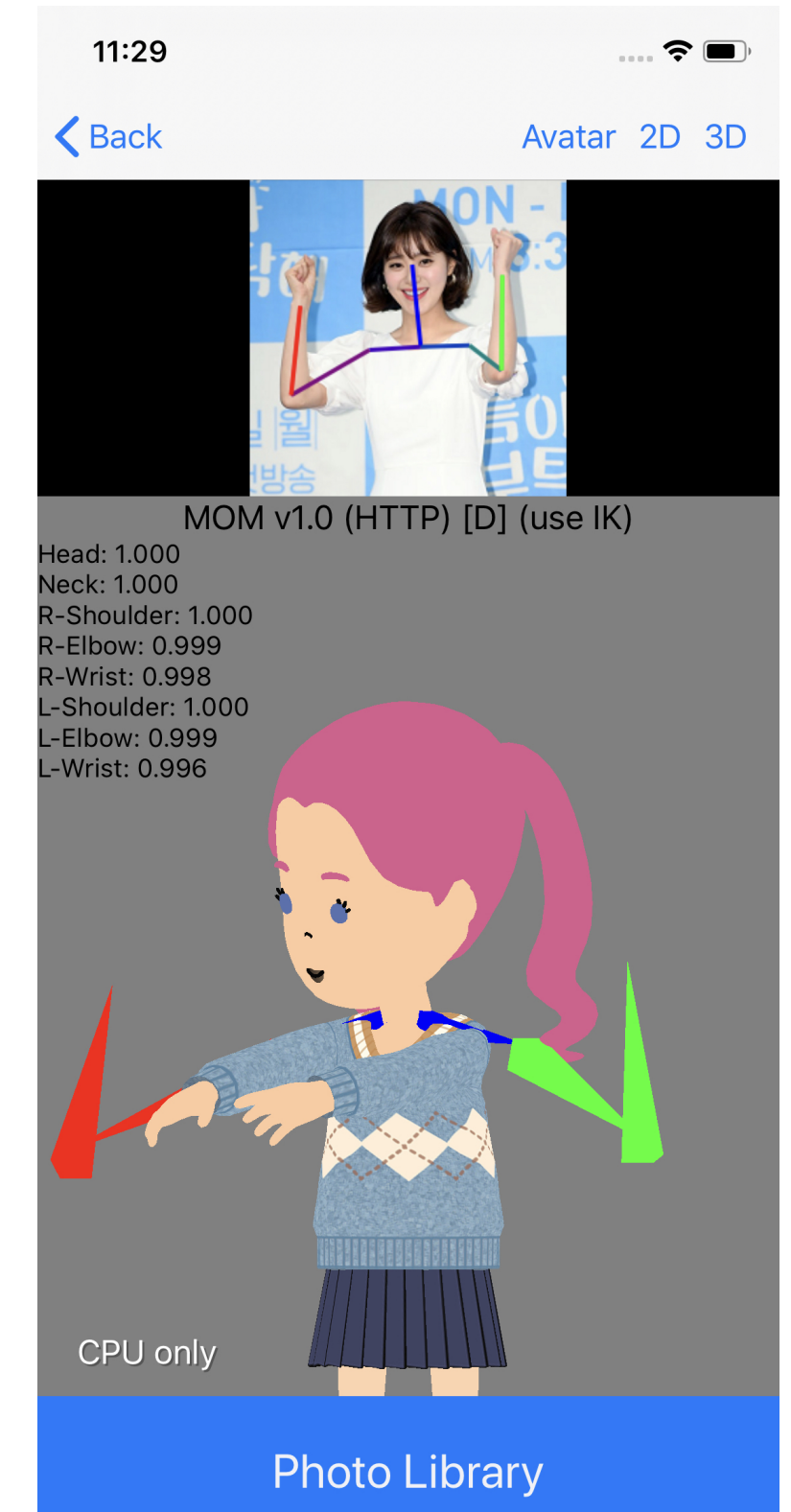
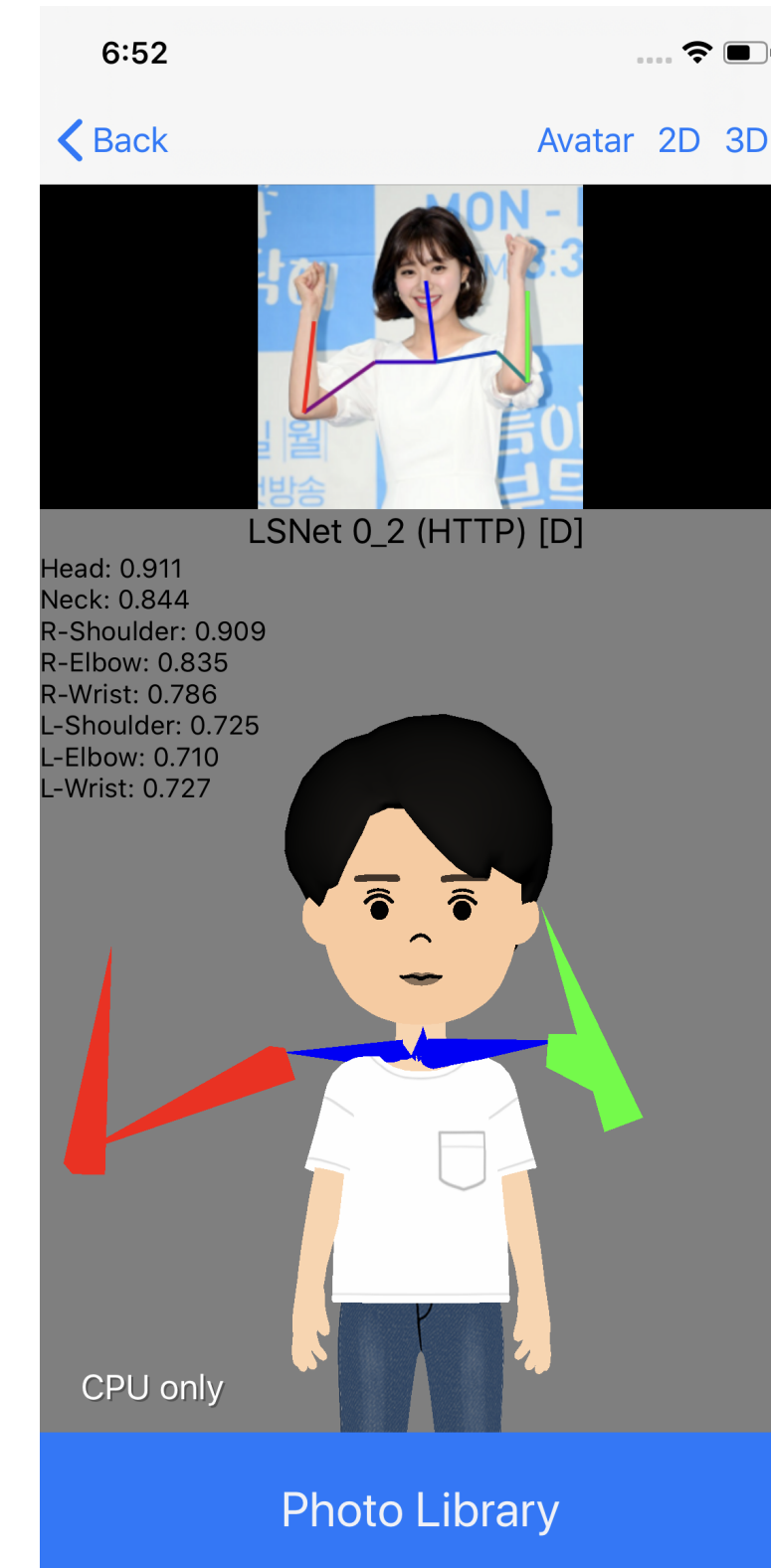
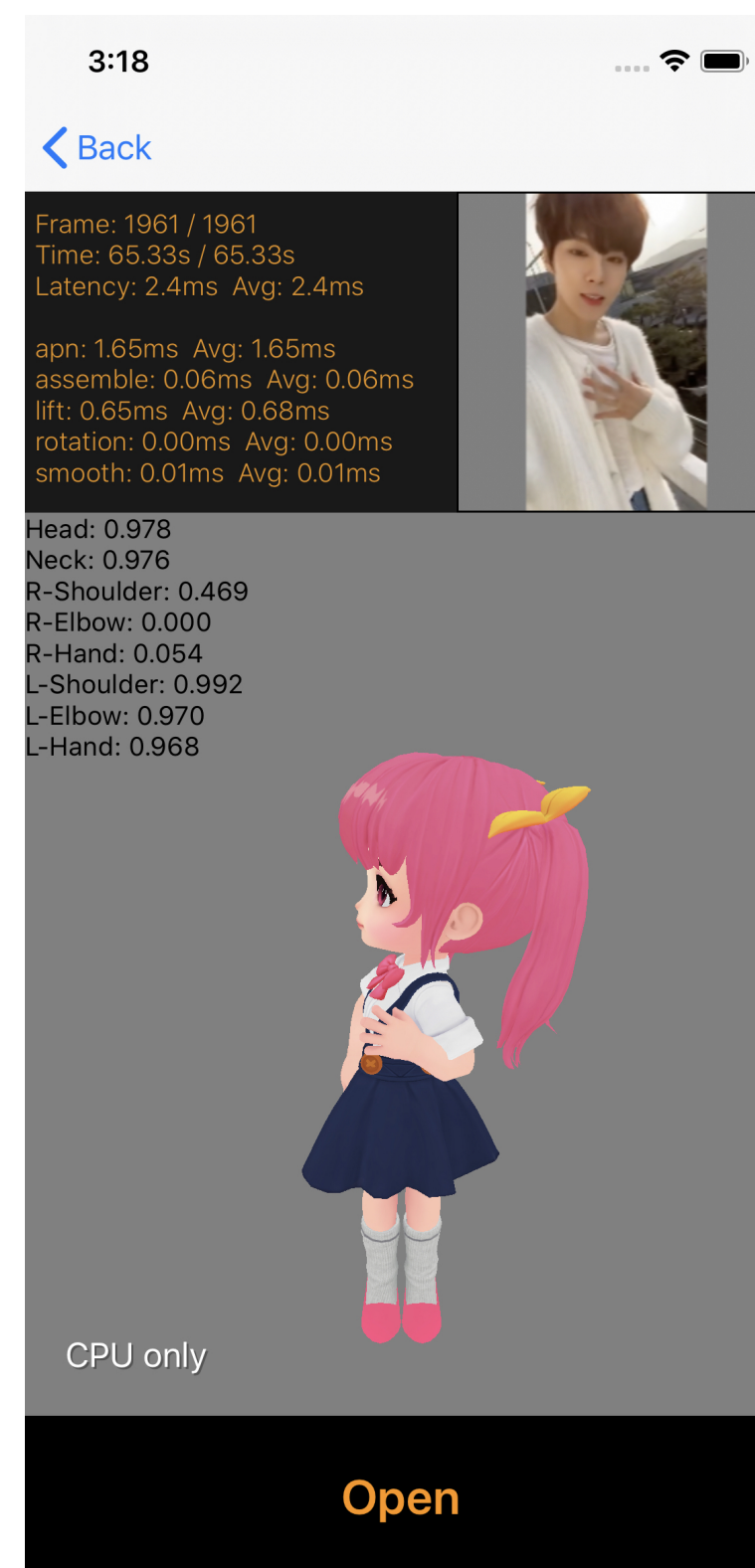
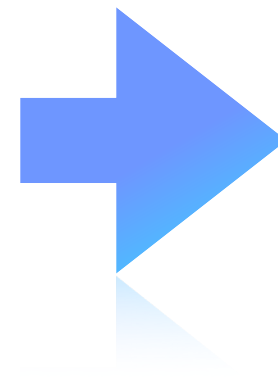
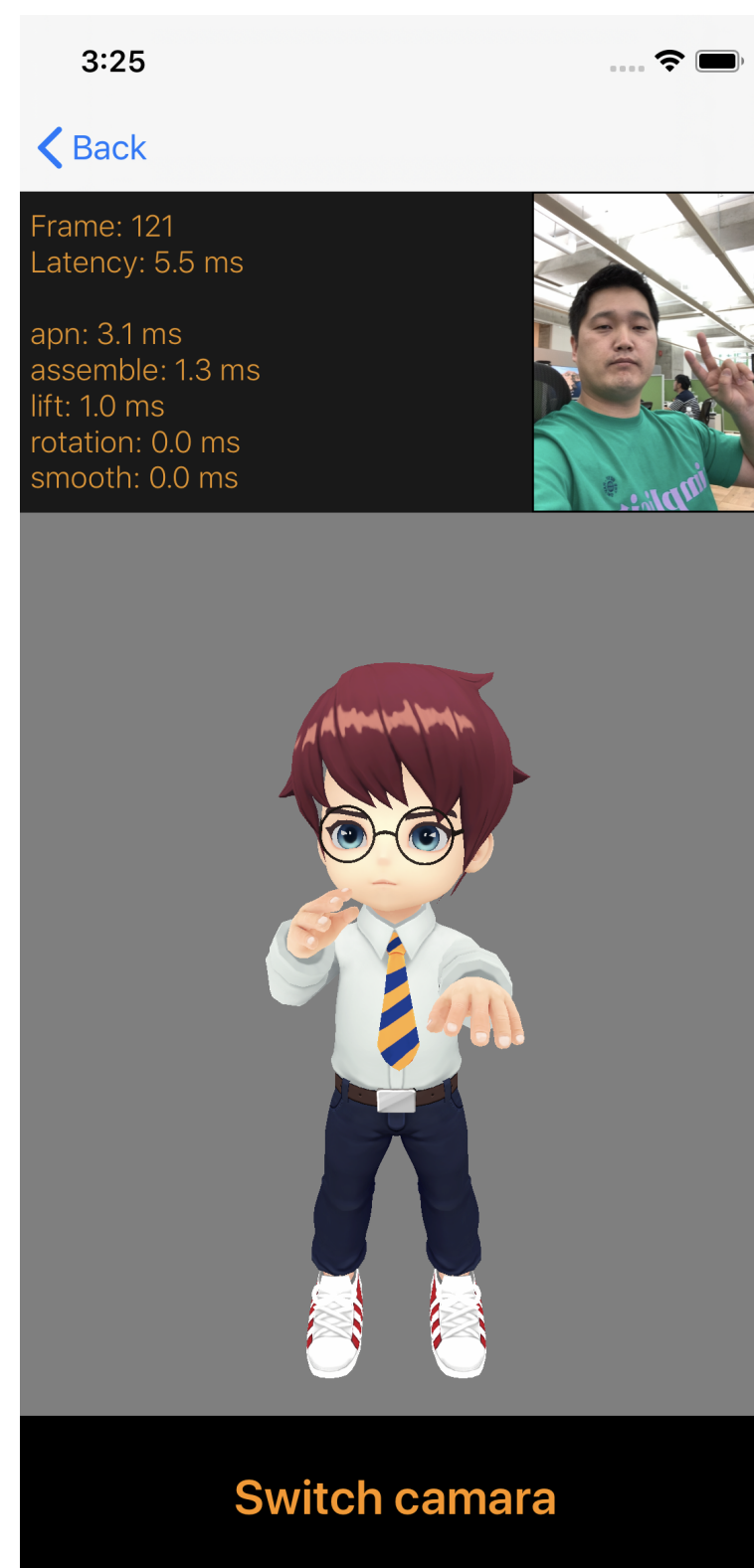
Retargeting with Quaternion



① `setRotation(type: LeftLowerArm, quaternion: qValue)`

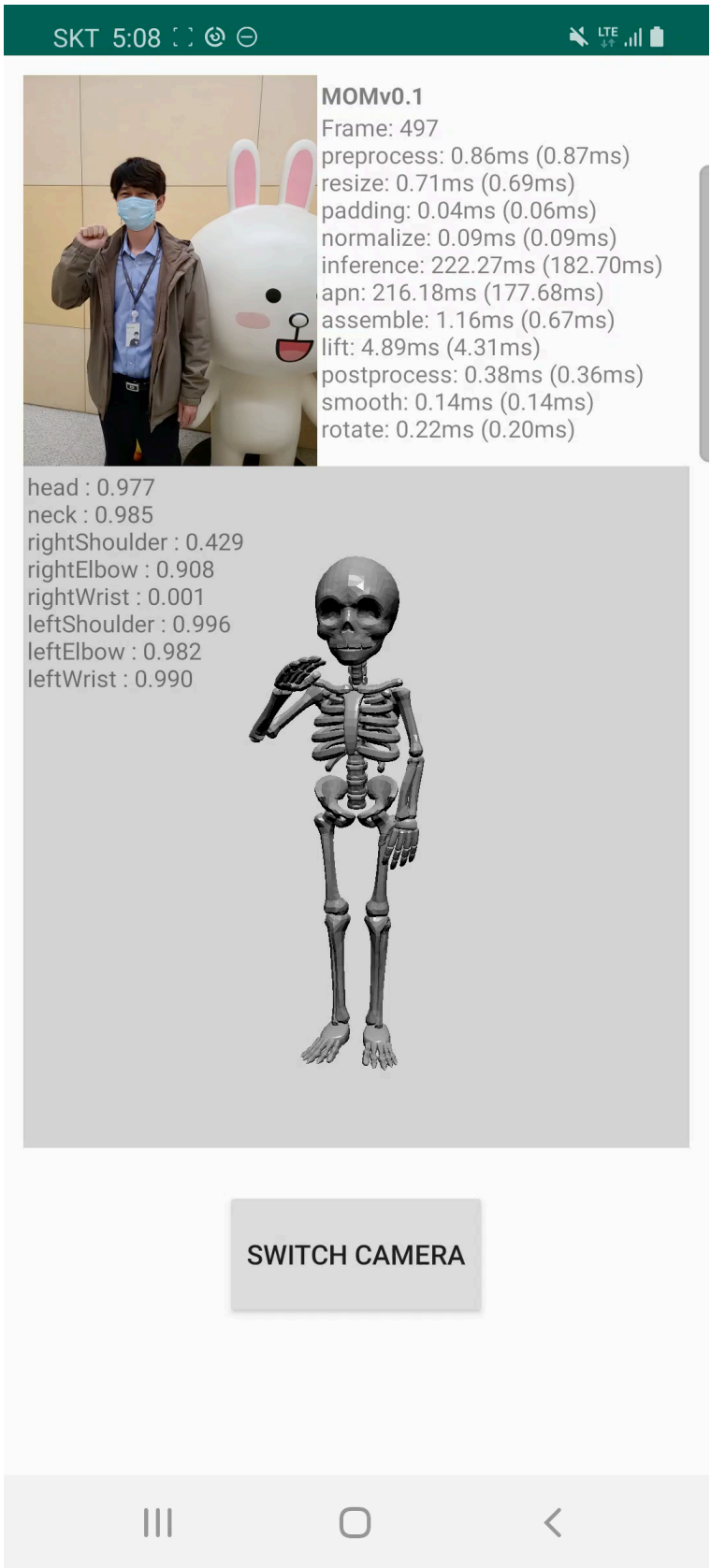
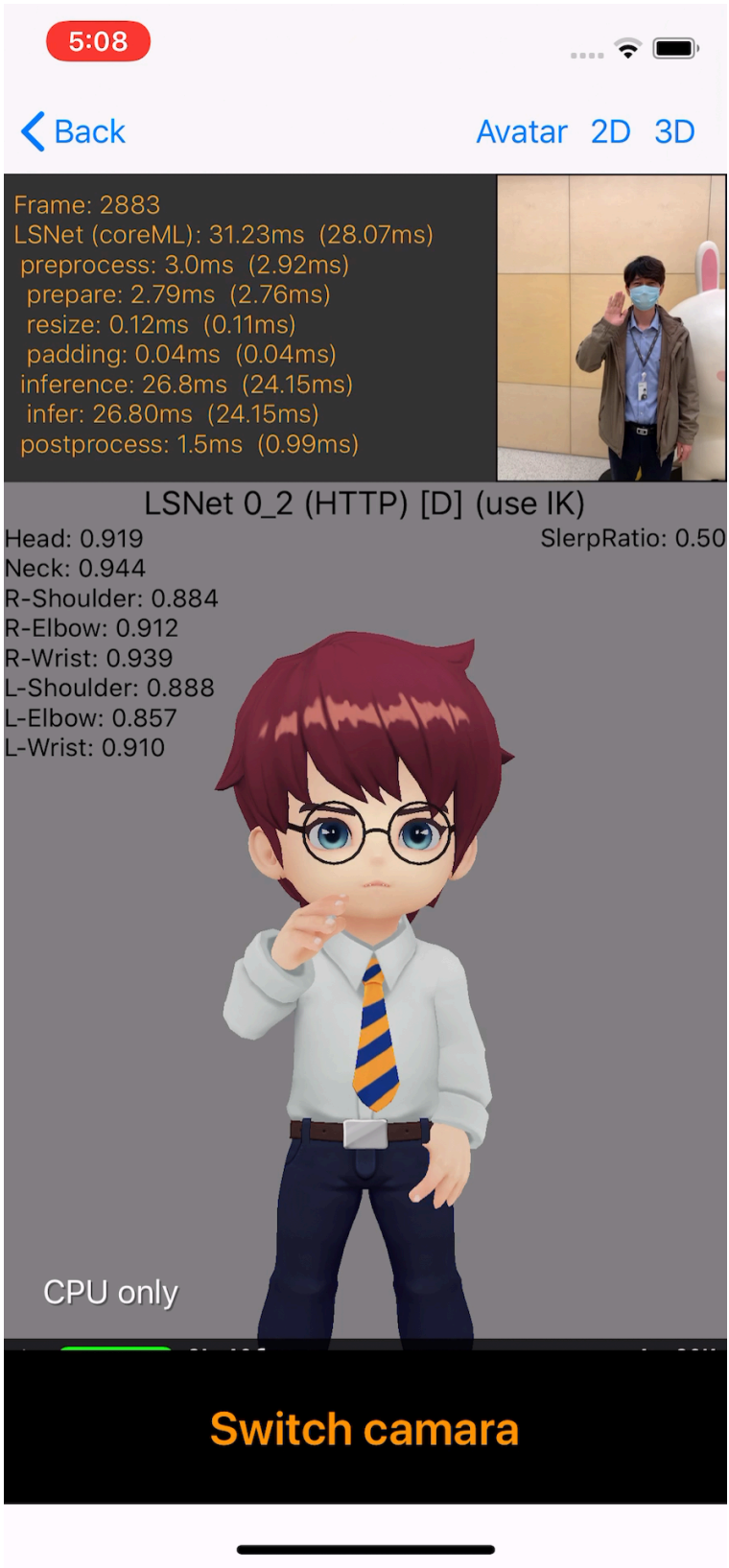
디버깅 환경 개발

- 2D/3D Skeleton 및 Confidence value 출력
- 어디가 문제인지 빠른 확인 가능, 문제의 범위를 좁힐 수 있음 → 개발 속도 향상



Demo Application

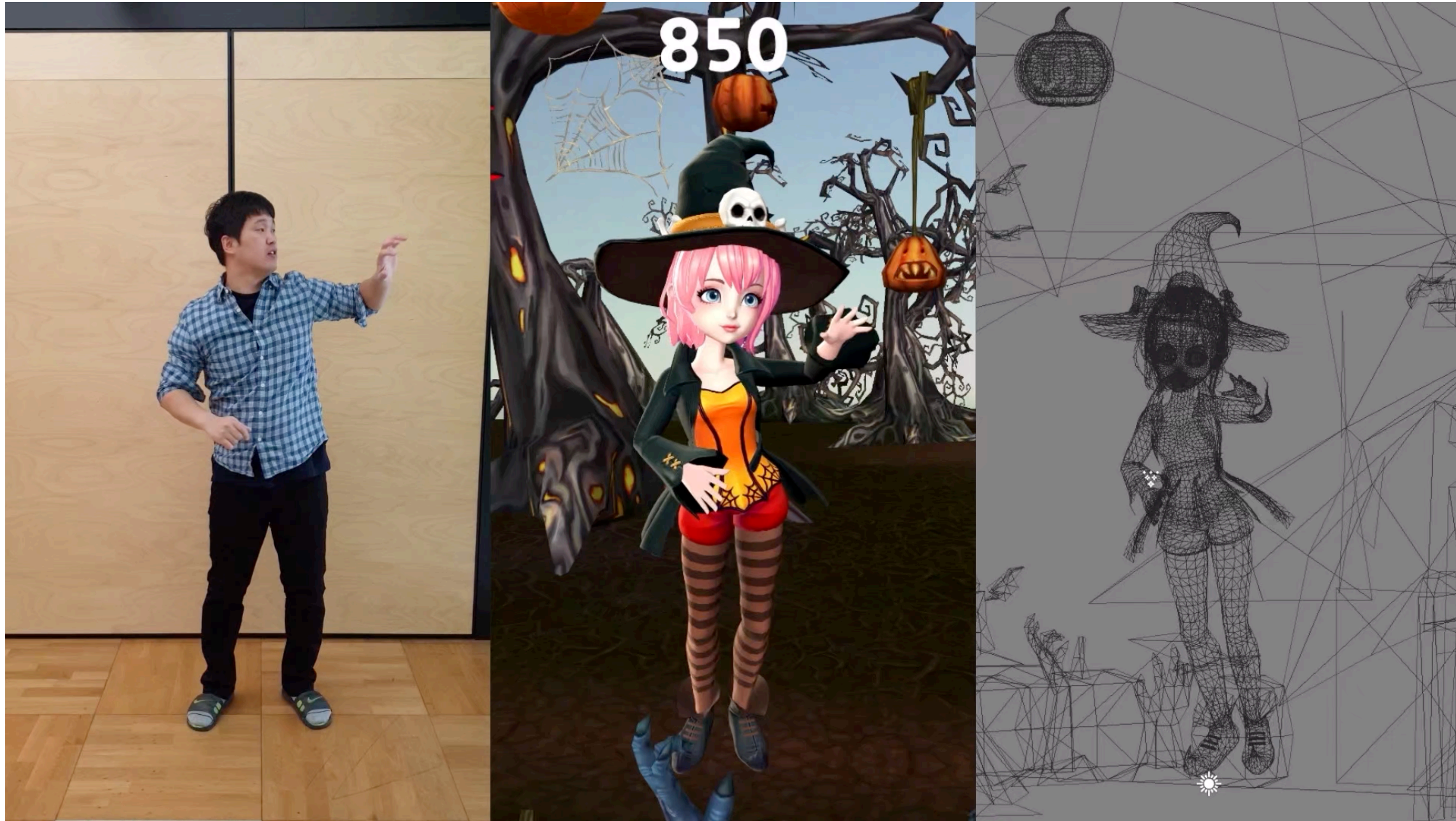
iOS and Android devices



Demo (2019년 가을, 1st Version)



Demo (2019년 겨울, ICCV, 2nd Version)



Field Test (2019년 겨울, ICCV CLOVA Booth)

DEVVIEW
2020



Field Test (2019년 겨울, LINE DEV DAY)

N DEVIEW
2020

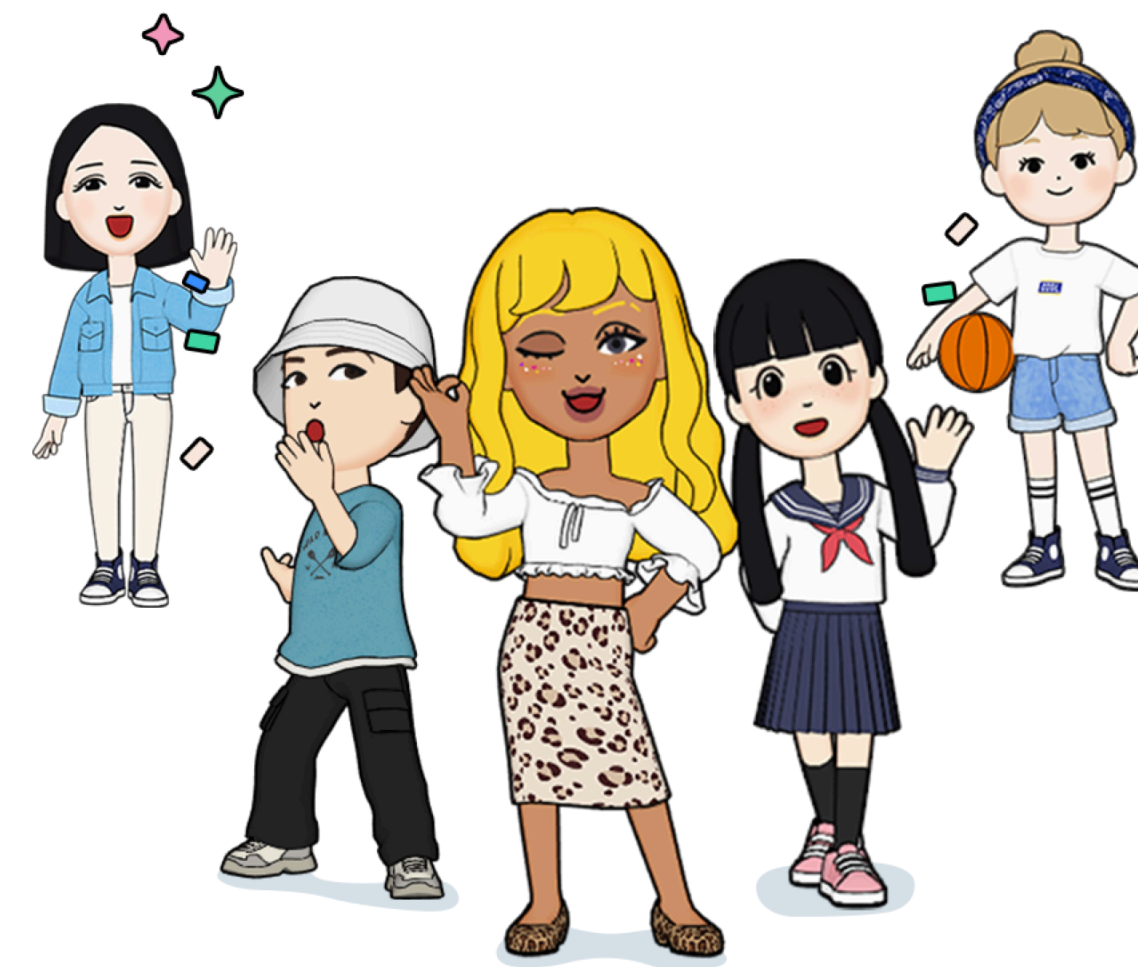
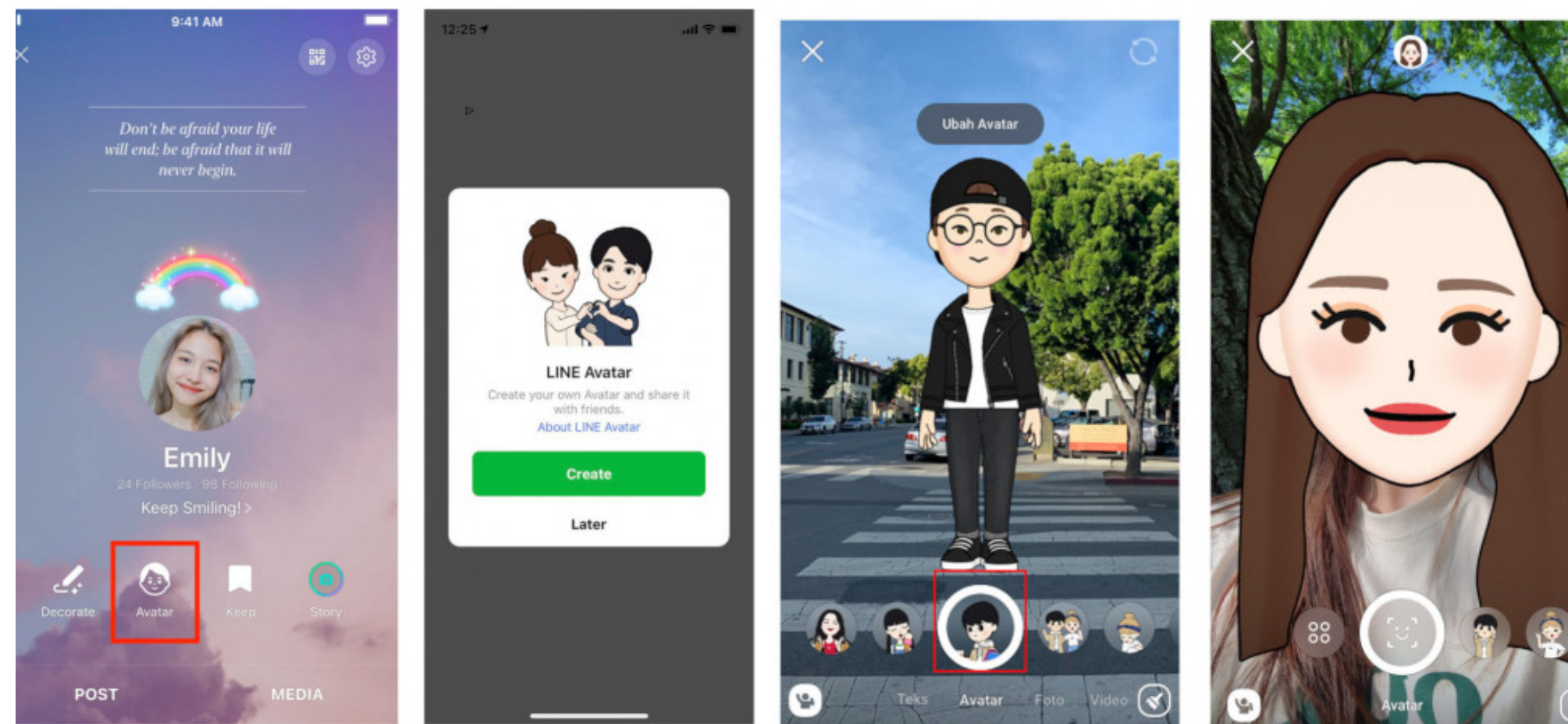


6. Optimization for Service

게임은 이제 시작!
배보다 배꼽이 더 크다!

상반신 특화 모델 개발

- LINE 메신저 아바타와 서비스 준비
- 영상 통화 특성상 혼자 촬영하기 때문에 전신보다는 상반신인 경우가 대부분



LINE 모바일 버전에서 확인해 주세요.
LINE 모바일 버전에서 QR 코드를 스캔하면 내용을 확인
할 수 있습니다.



상반신 특화 모델 개발

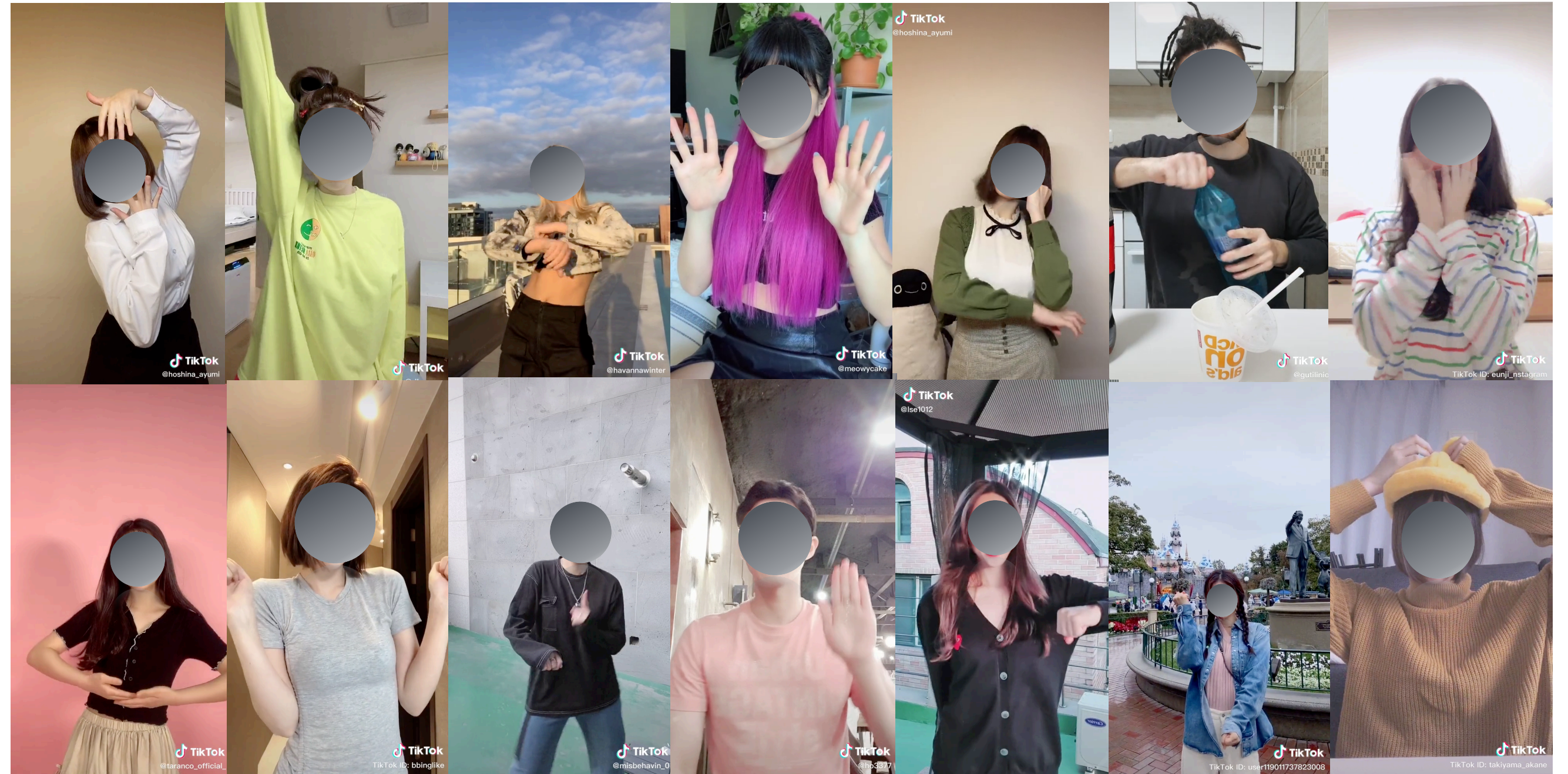
상반신에 특화된 더 정확하고 빠른 모델 개발 (MPJPE: 45mm)

Inference Time

Device	iPhone7		iPhone 11 Pro	
Mode	CPU	GPU	CPU	GPU
MS	21	17	11	9
FPS	47.62	58.82	90.91	111.11

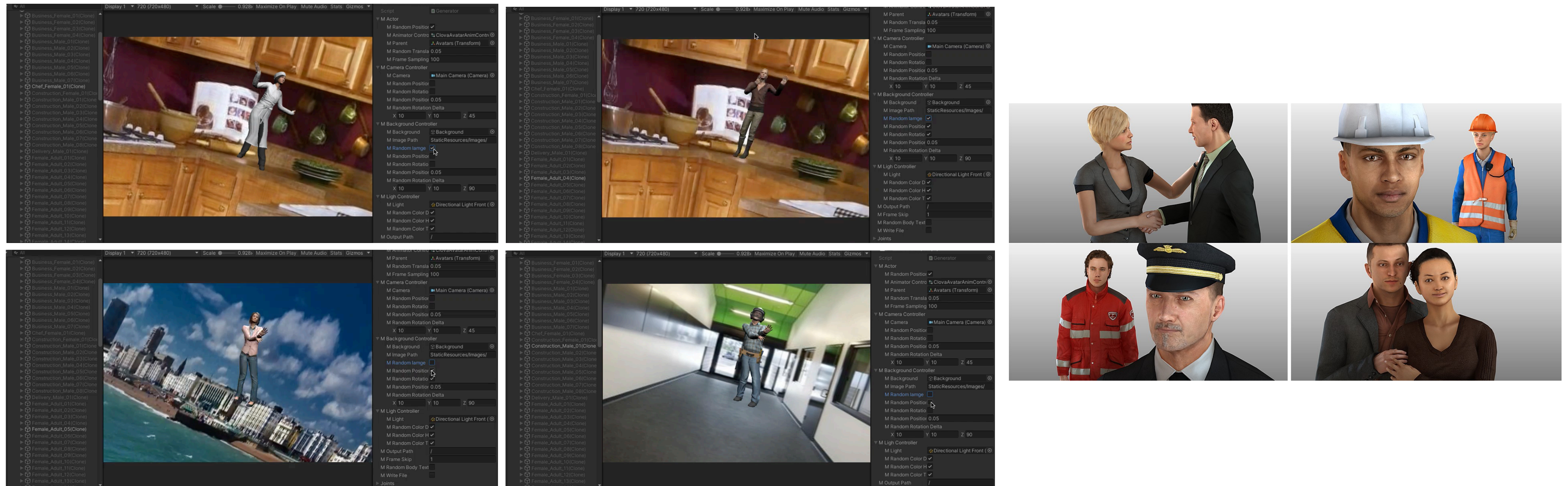
서비스에 특화된 합성 데이터셋 만들기

- Animation: Capture using Perception Neuron
- 3D Character: 115 Characters of Microsoft Rocketbox



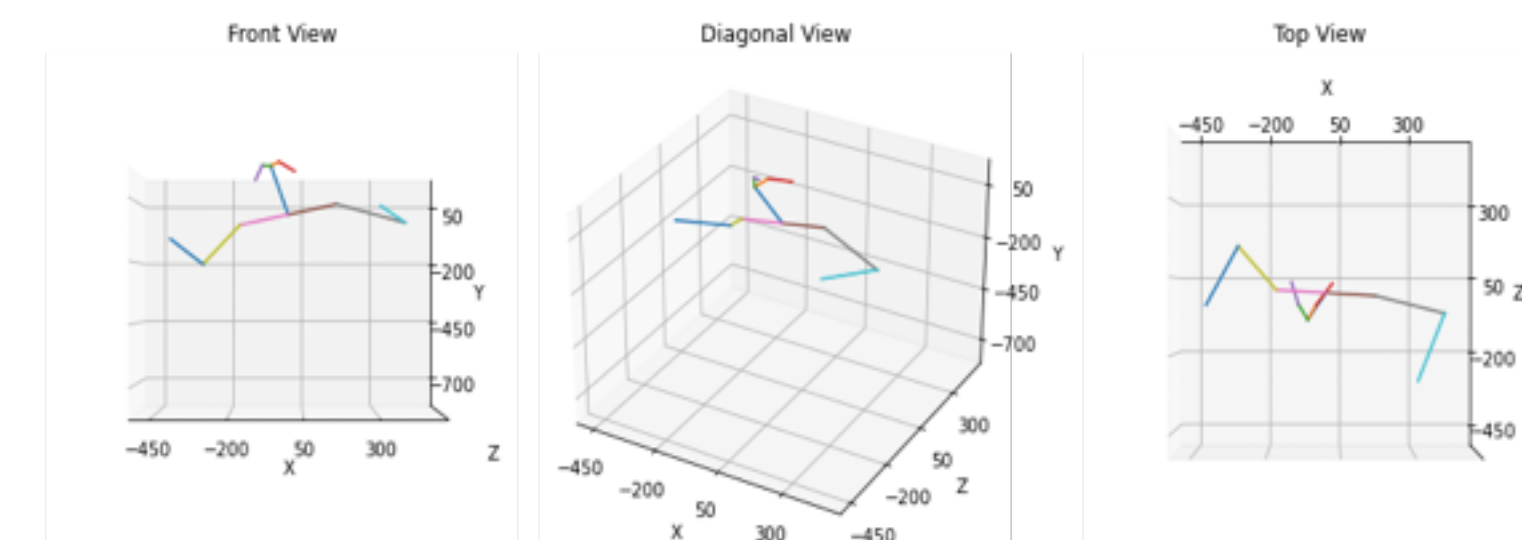
Synthesize using Unity

Transform with Character, Background, Light and Camera



Synthesize using Unity

원하는 동작의 합성 데이터셋 자체 확보



보다 더 자연스럽게 자세 보정하기

IK 5-DOF Rotation Constraints를 적용한 자세 보정

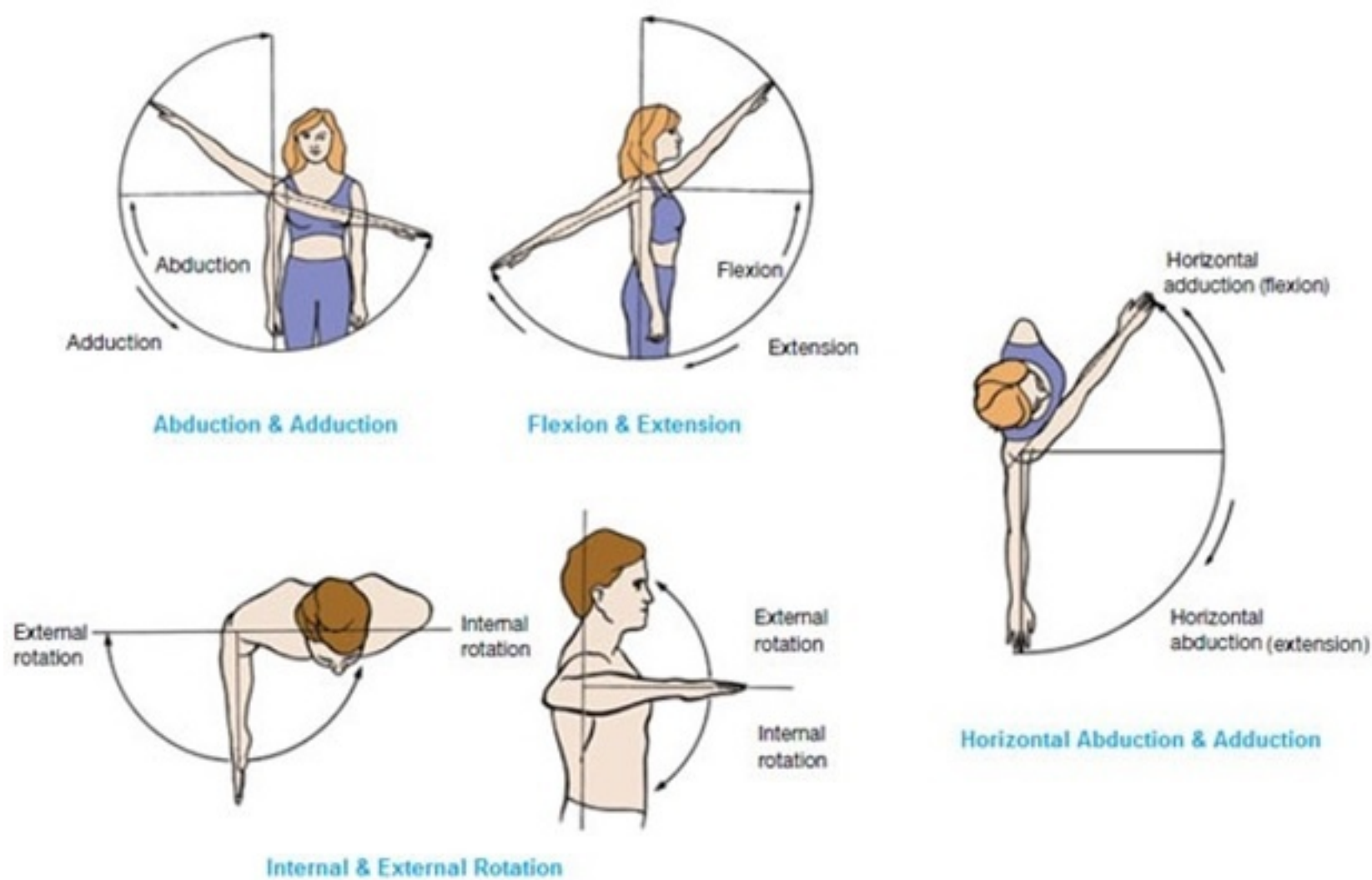


Table 1. Motion range of human joints.

Motion	Range
Shoulder flexion (Flex.)/extension (Ext.) (Z_2)	135°/50°
Shoulder adduction (Add.)/abduction (Abd.) (Z_3)	108°/0°
Shoulder internal (Int.)/external-rotation (Ext. Rot) (Z_1)	90°/90°
Elbow flexion (Flex.)/extension (Ext.) (Z_4)	0°/145
Palmar flexion/dorsiflexion (Z_6)	90°/70°
Ulnar deviation/radial deviation (Z_7)	55°/25°
Wrist pronation (Pron.)/supination (Sup.) (Z_5)	90°/90°

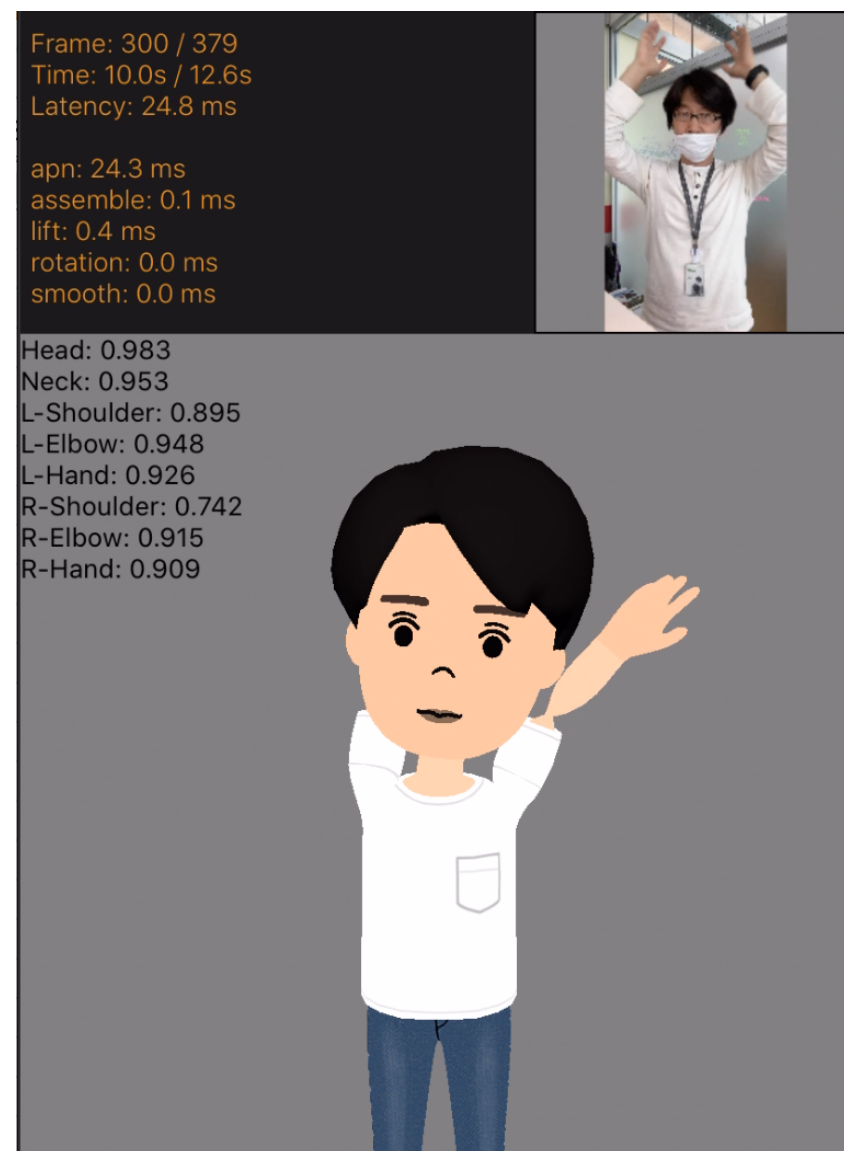
Table 2. Motion range of 5-DOF upper-limb exoskeleton.

Motion	Range
Shoulder Flex./Ext. (θ_2^5)	135°/15°
Shoulder Add./Abd. (θ_1^5)	90°/30°
Shoulder Int./Ext. Rot (θ_3^5)	90°/30°
Elbow Flex./Ext. (θ_4^5)	118°/0°
Elbow Pron./Sup. (θ_5^5)	30°/30°

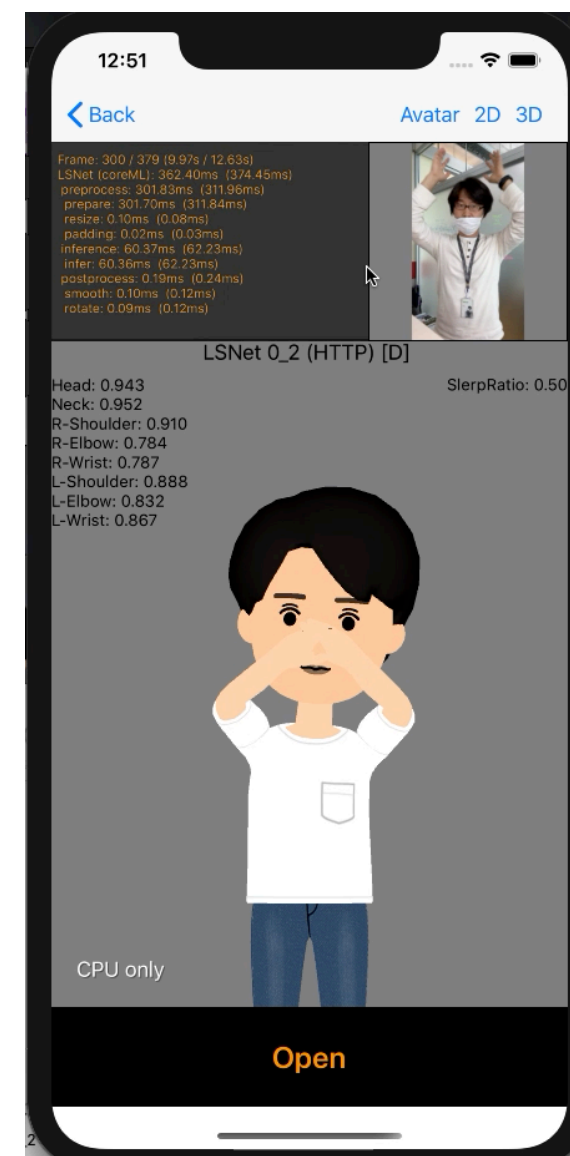
7. QA Results

모델 정확도 개선

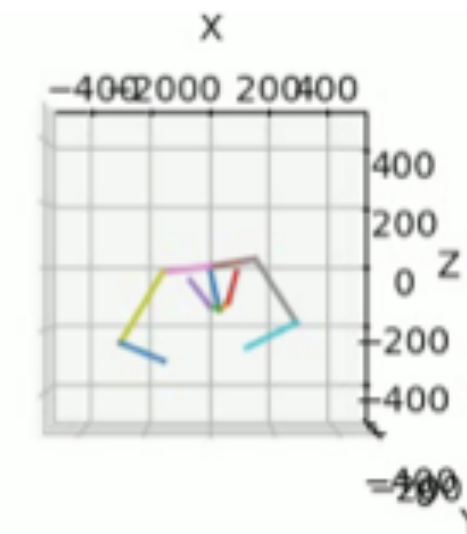
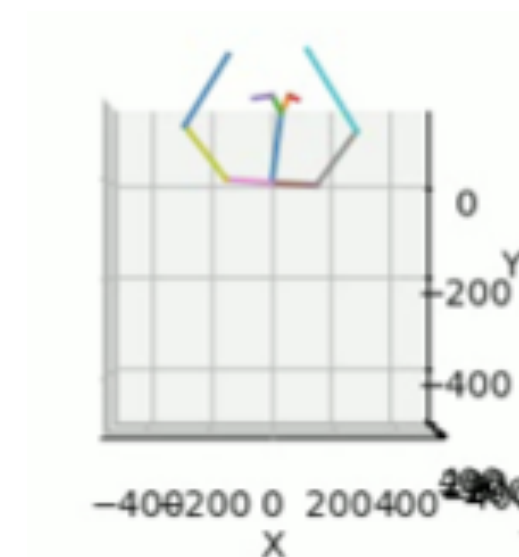
- 마스크 착용, 마스크와 상의 모두 흰색 → Dataset에는 없는 보기 드문 이미지
- 양쪽 팔꿈치가 반정도 안보이는 경우



개선 전

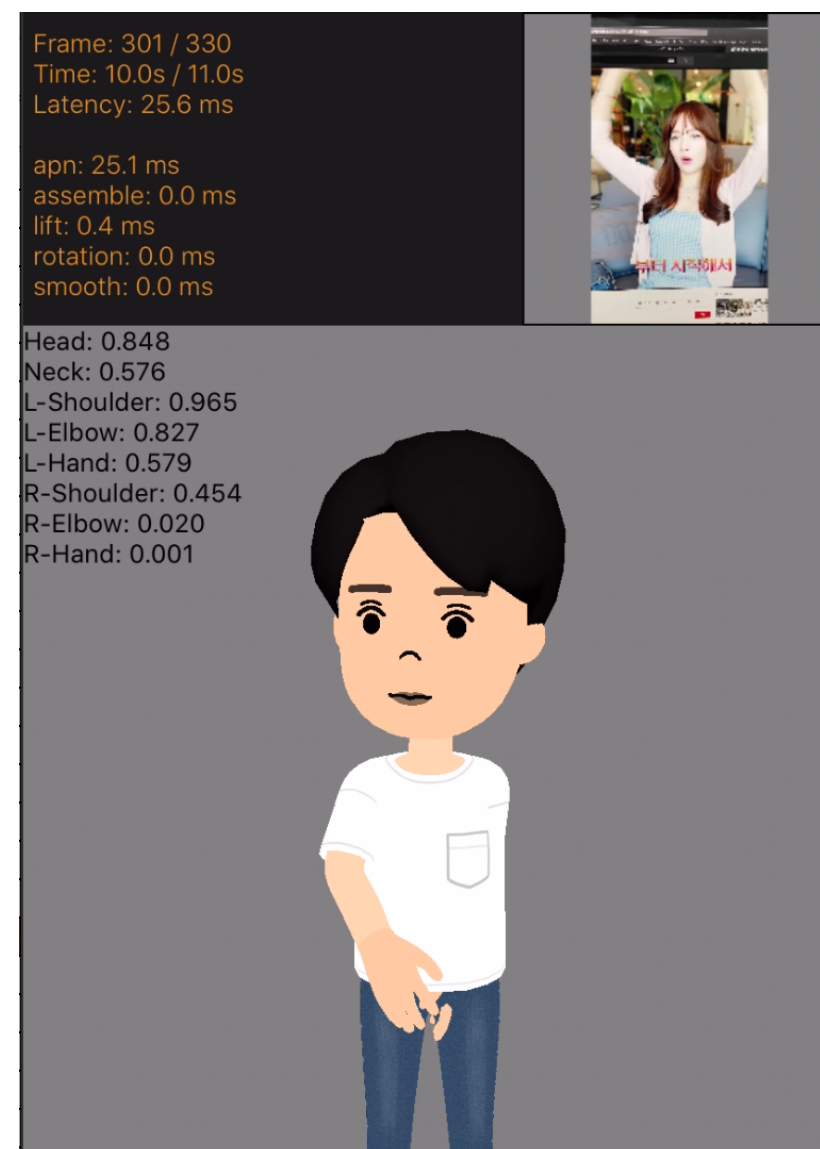


개선 후

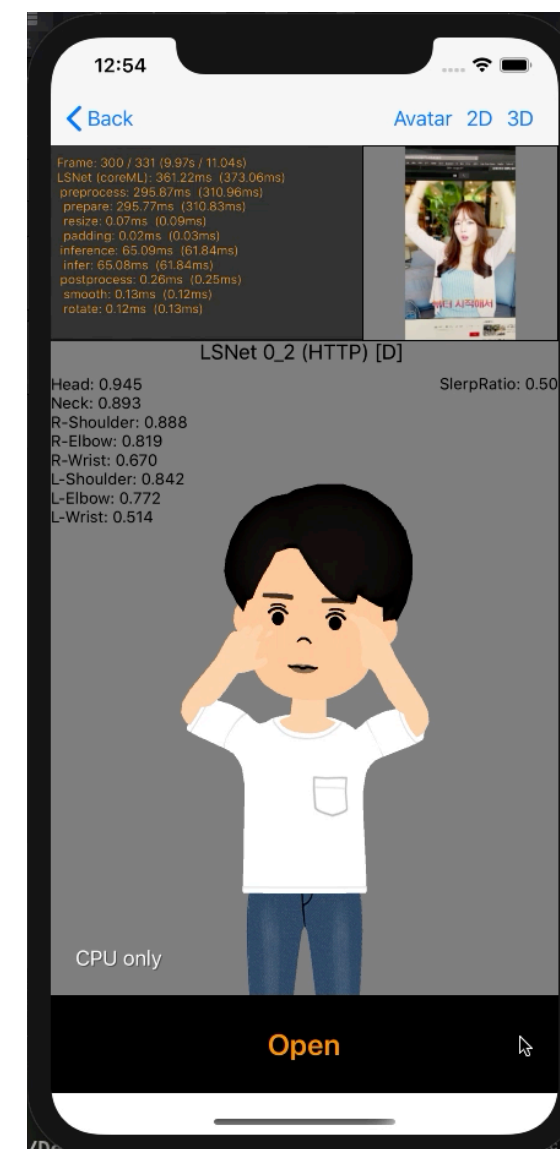


모델 정확도 개선

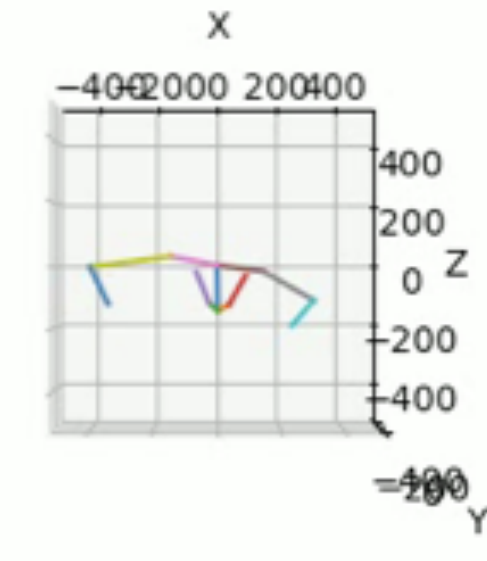
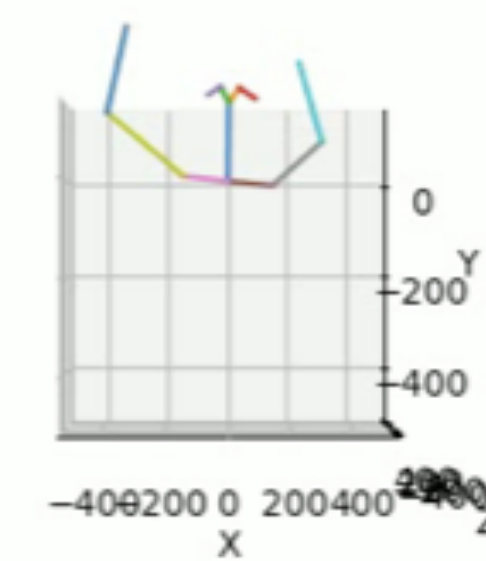
- 양손이 이미지에 없는 경우
- 긴머리가 양쪽 어깨를 가린 경우



개선 전



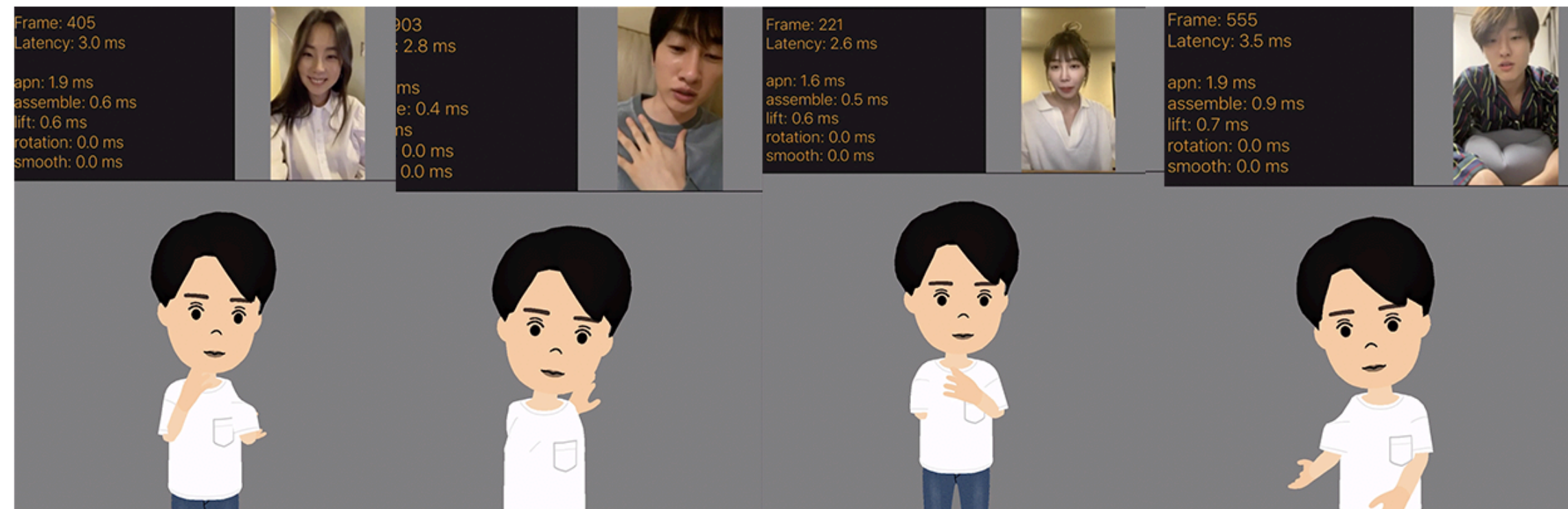
개선 후



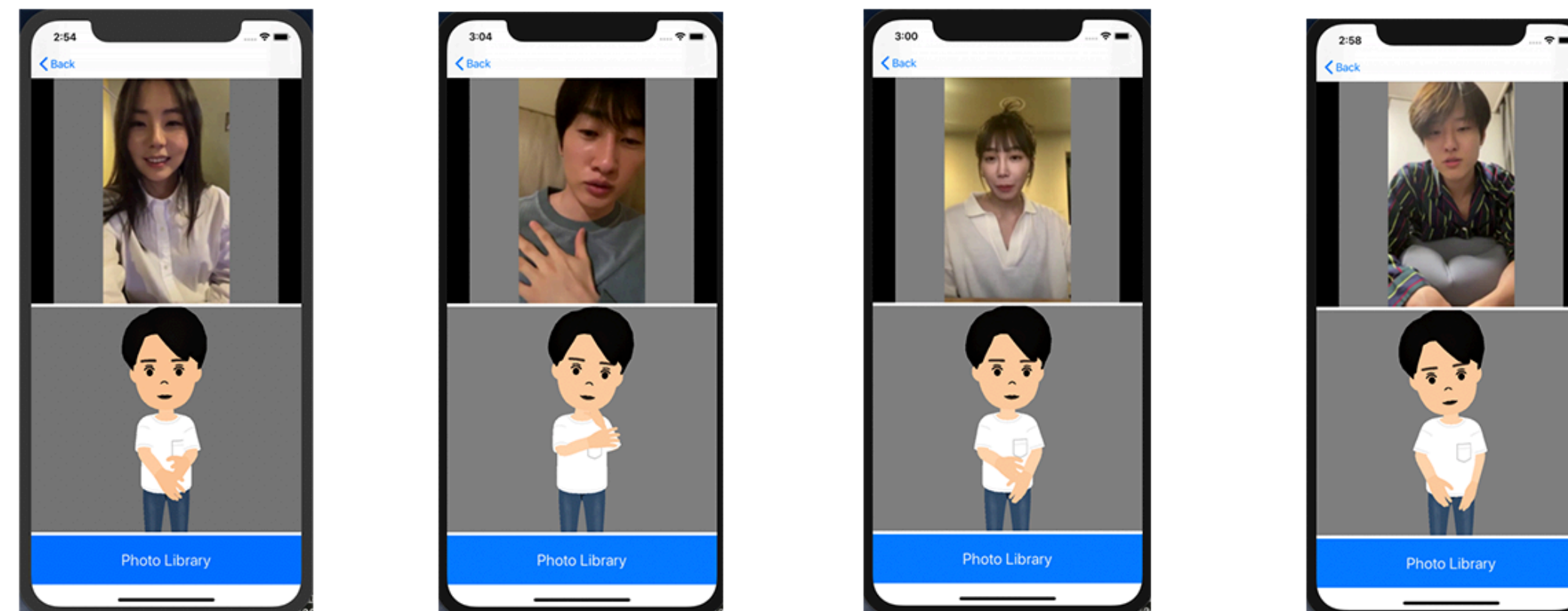
후처리 모듈 개선

- 양손 혹은 한쪽 손이 이미지에 없는 경우
- 양쪽 어깨가 이미지에 없거나, 양쪽 팔꿈치가 이미지에 없는 경우

Before

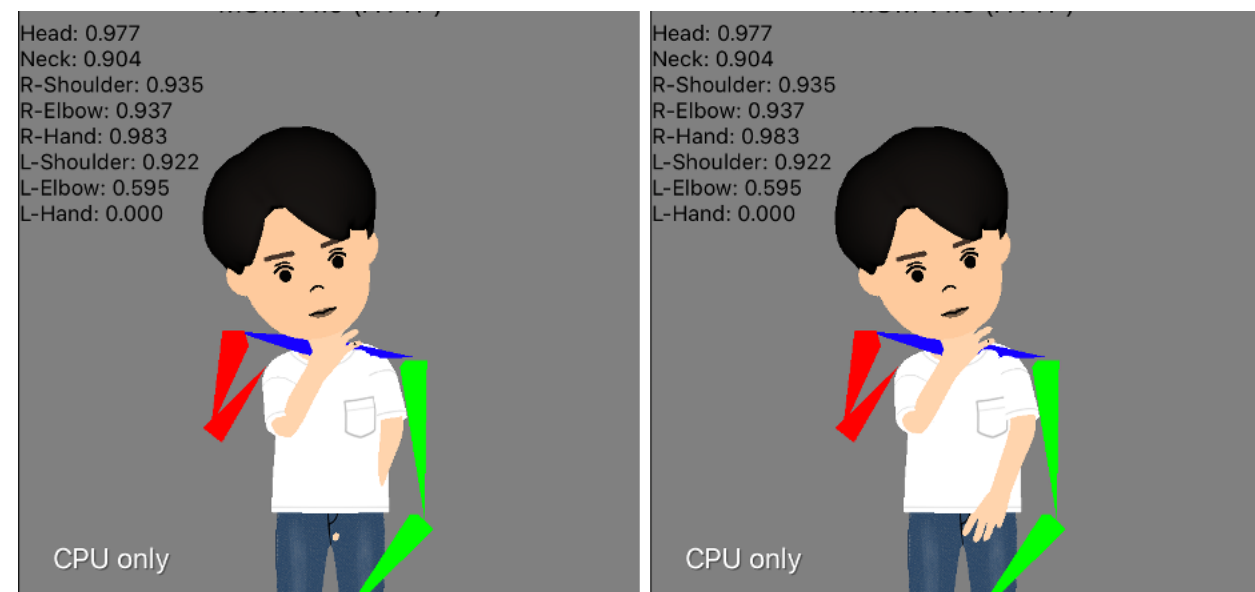


After



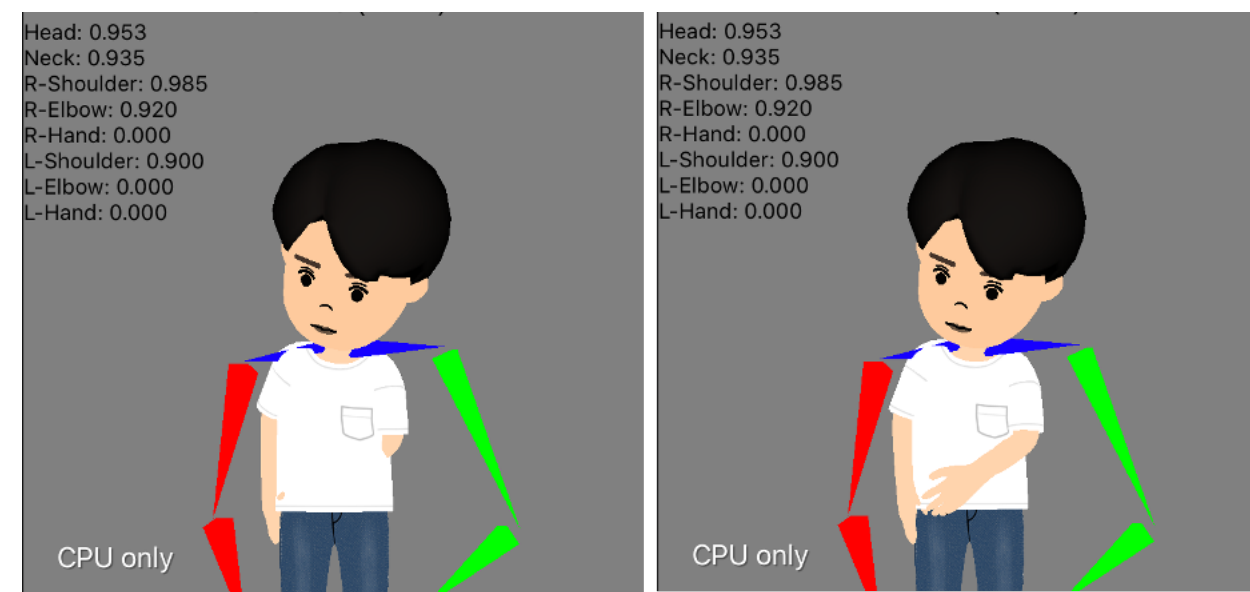
후처리 모듈 개선

- 팔이 몸통을 관통하는 경우
- 어깨가 꼬이거나, 몸통이 뒤틀어지는 경우



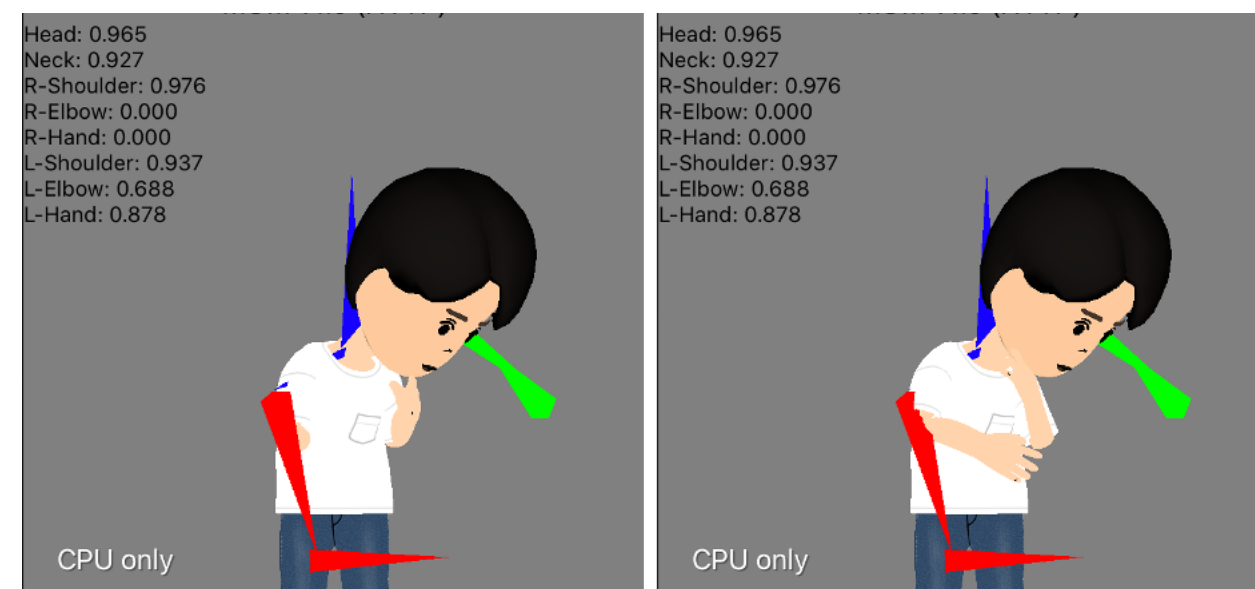
Original result

Using IK module



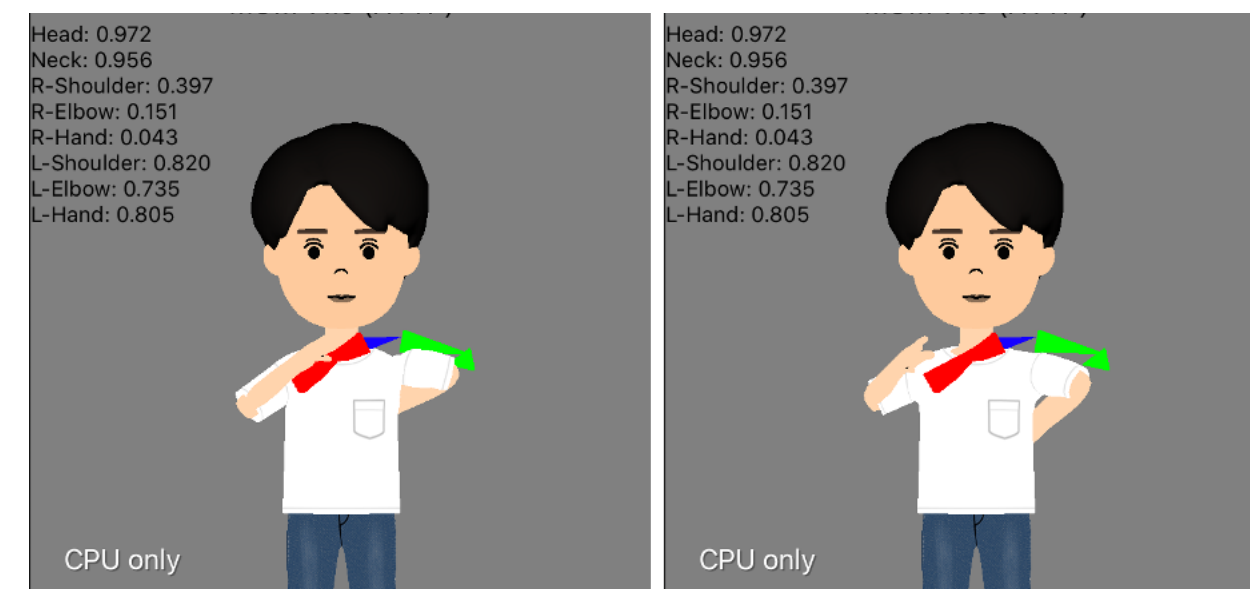
Original result

Using IK module



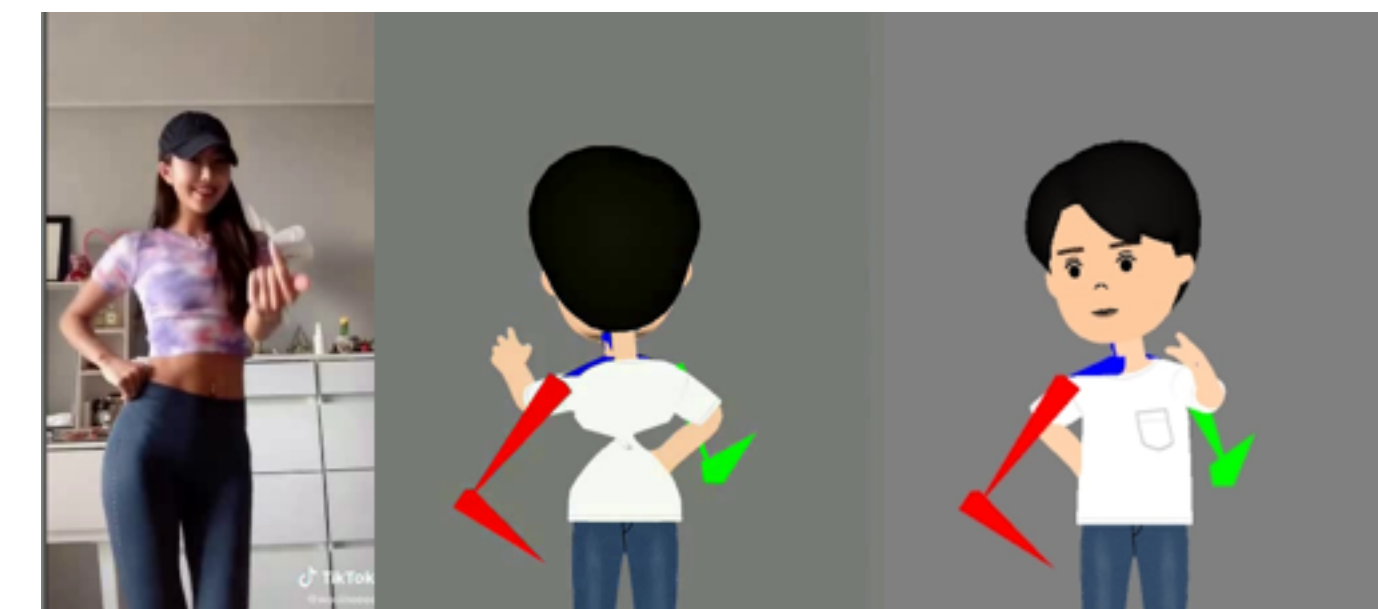
Original result

Using IK module



Original result

Using IK module



Original result

Using IK module

평가 점수 향상

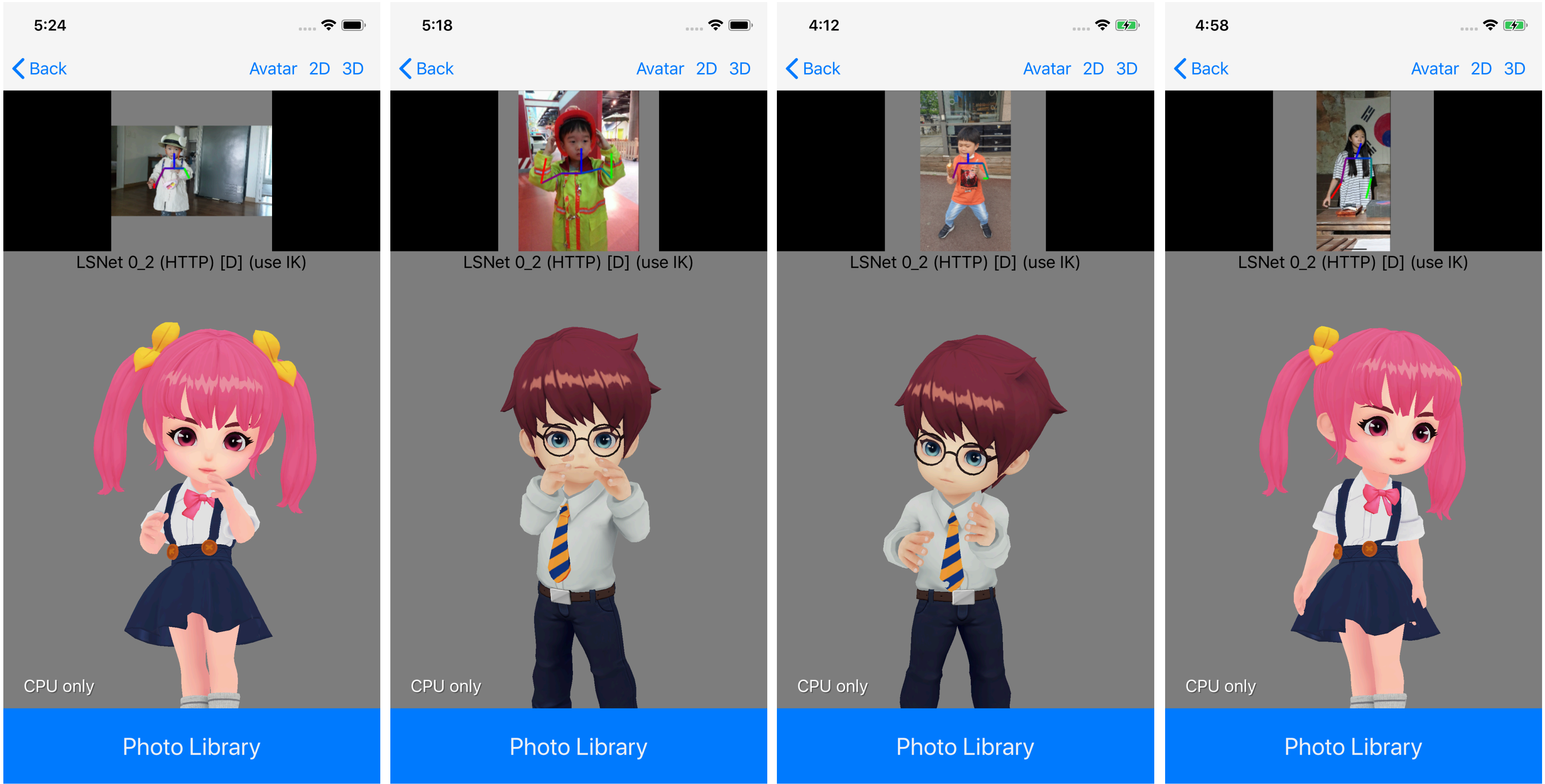
Performed by NAVER QA Team

1st QA: 86.14 (July, 2020)

2nd QA: 93.01% (October, 2020) +6.87% Up ↑

Demo

Human Scale



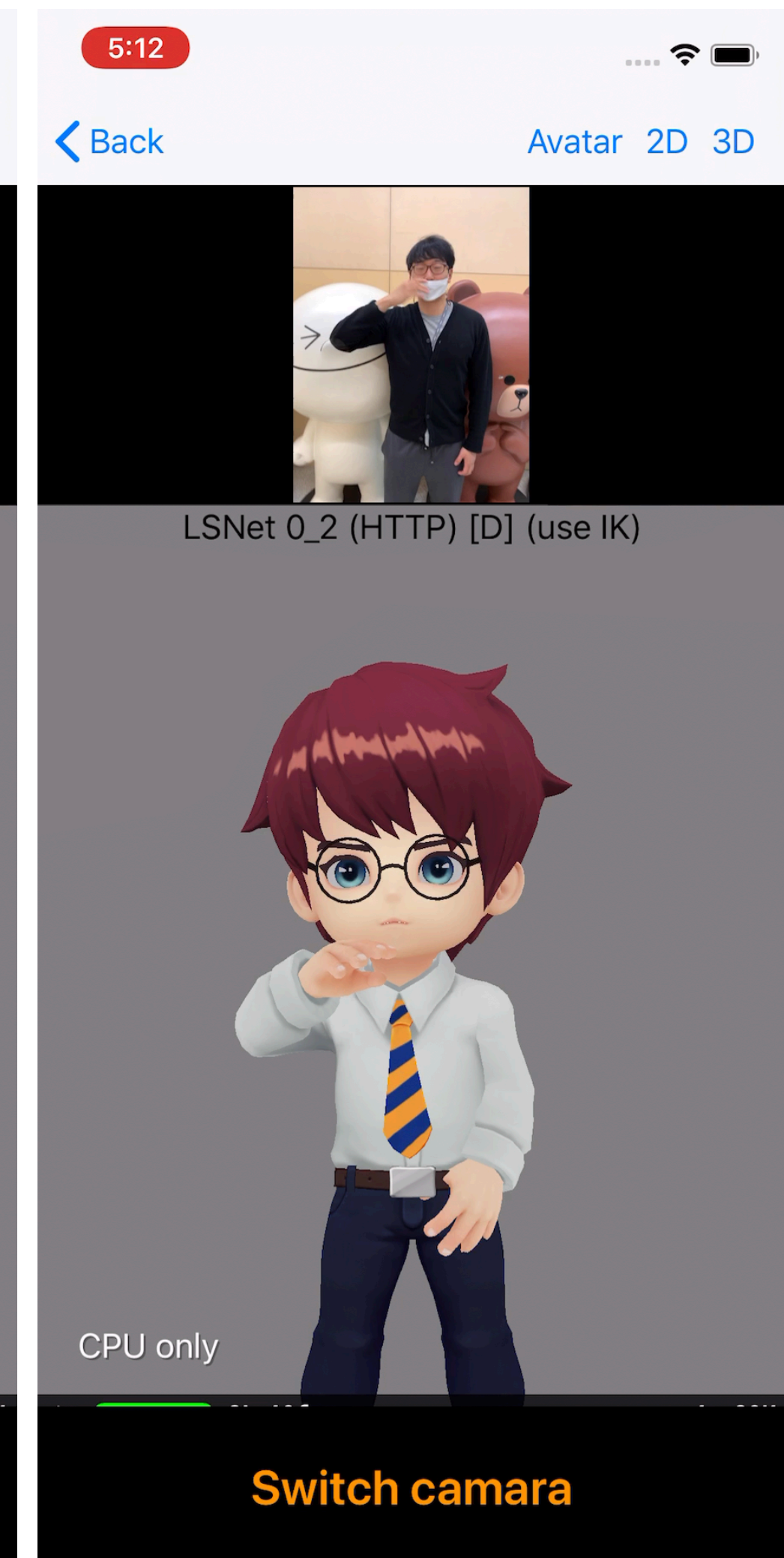
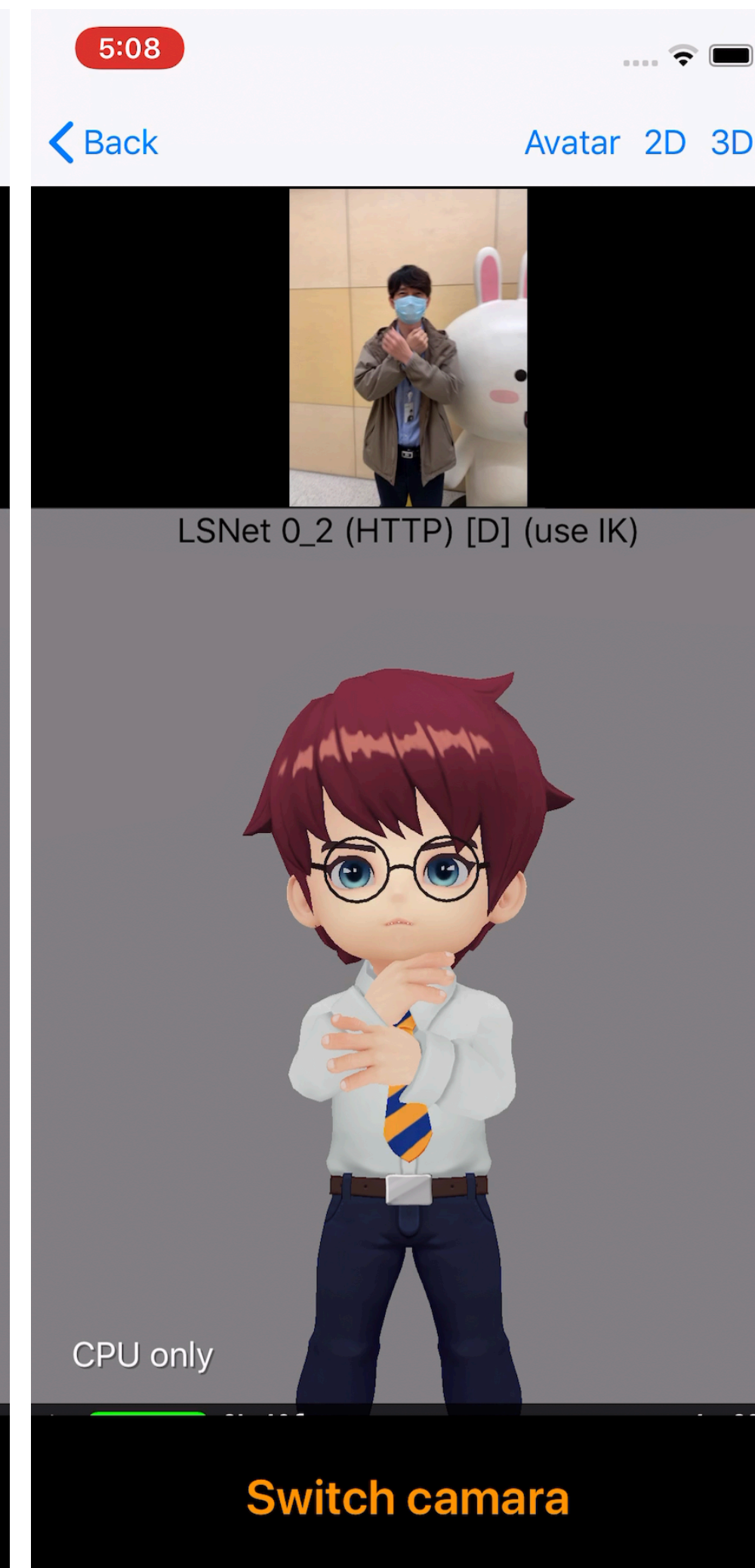
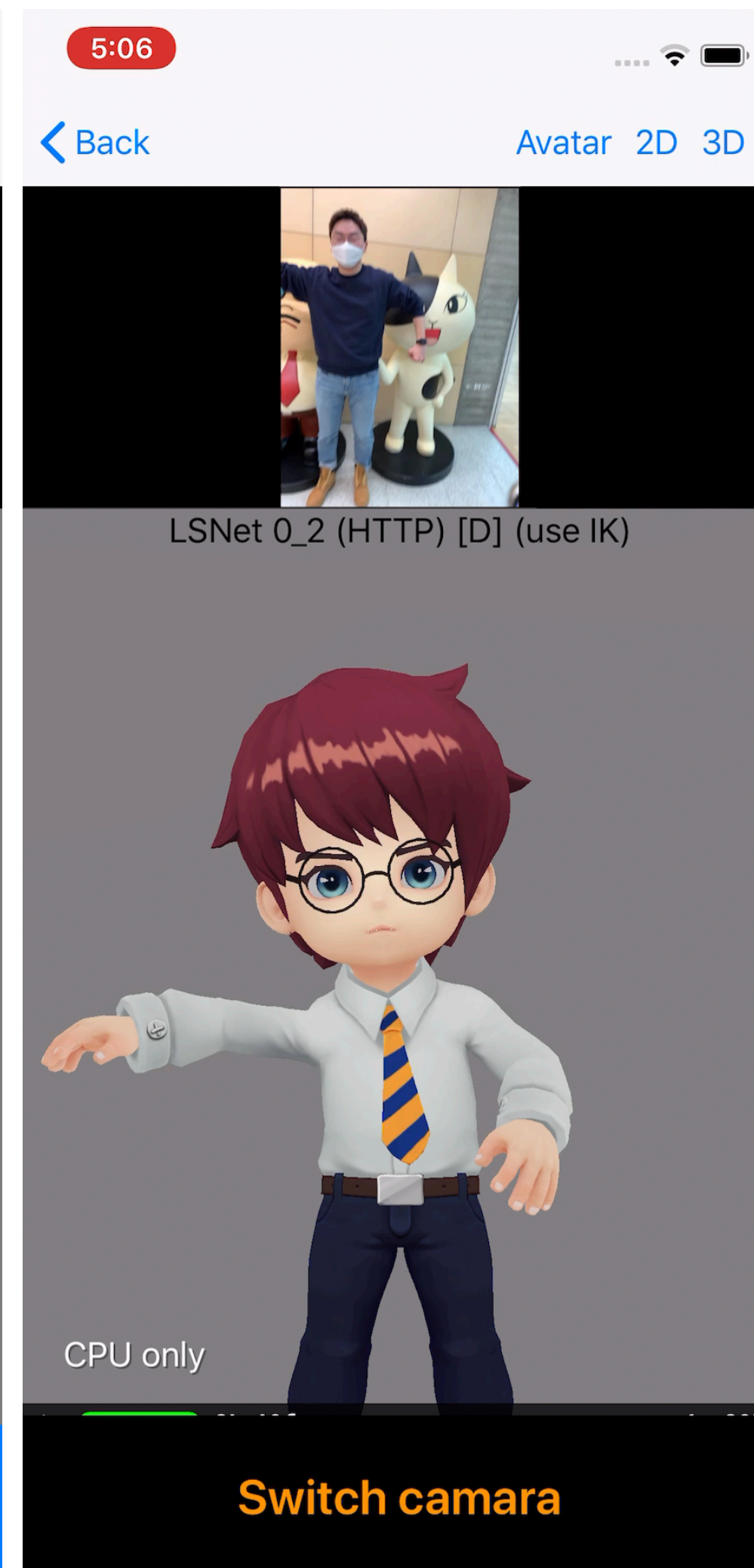
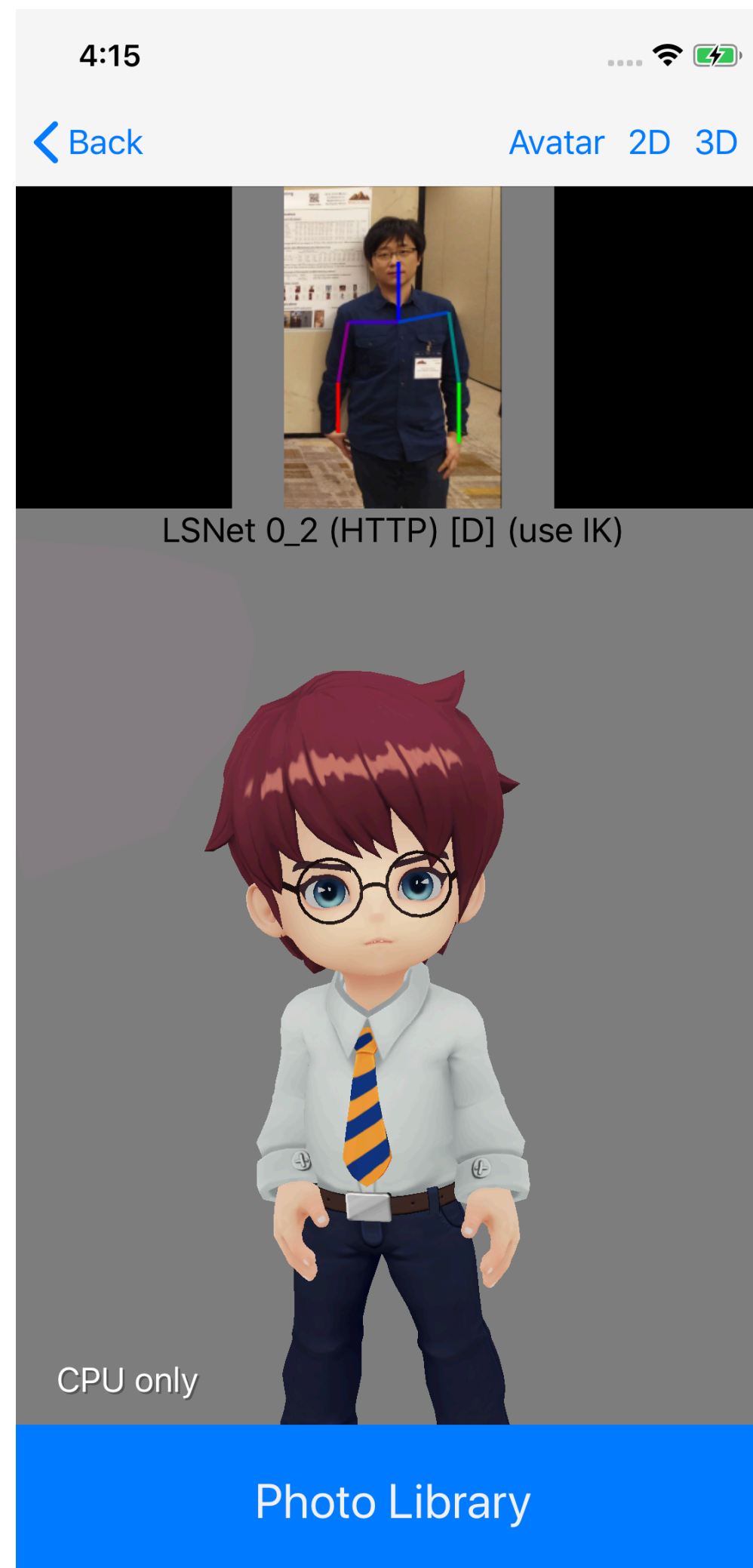
2 세

4 세

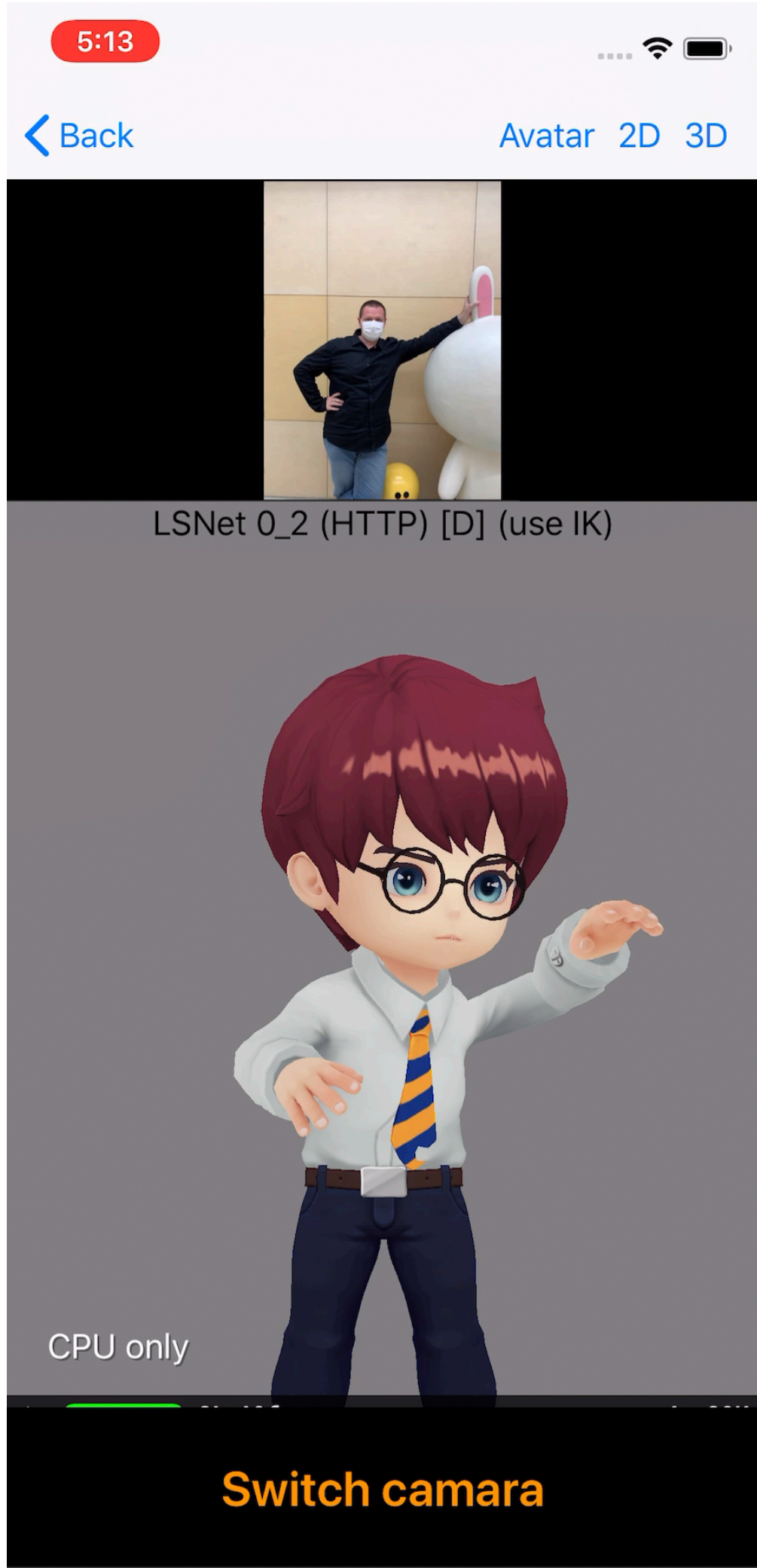
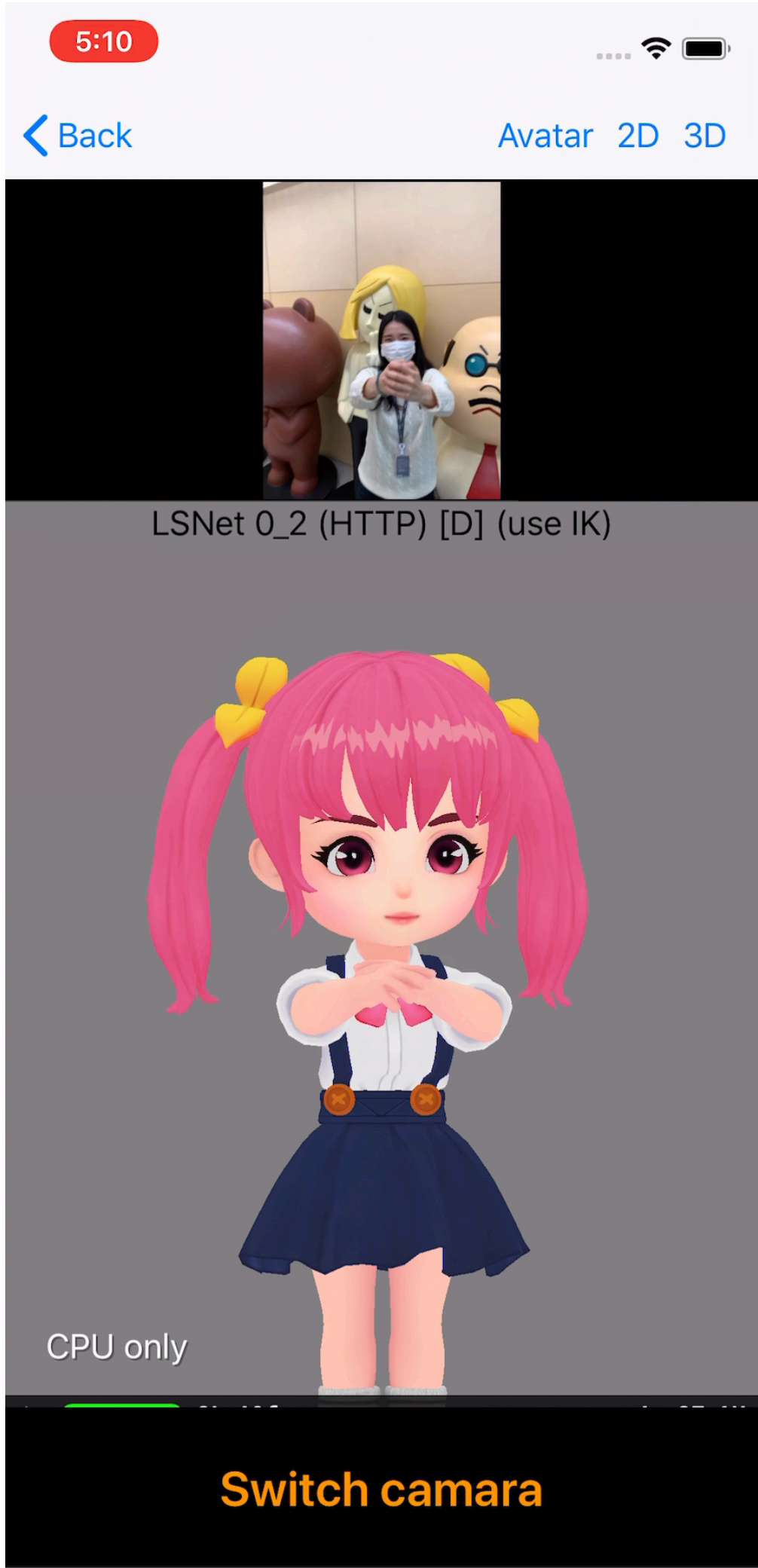
6 세

13 세

Various Pose with Mask



Different Characters



마지막으로 돌이켜 보면..



April, 2019

PC (GeForce 1060 Ti) + Web cam + Free sample 3D character
Model size: 60Mb, 12FPS (Render + Pre/Inference/Post)

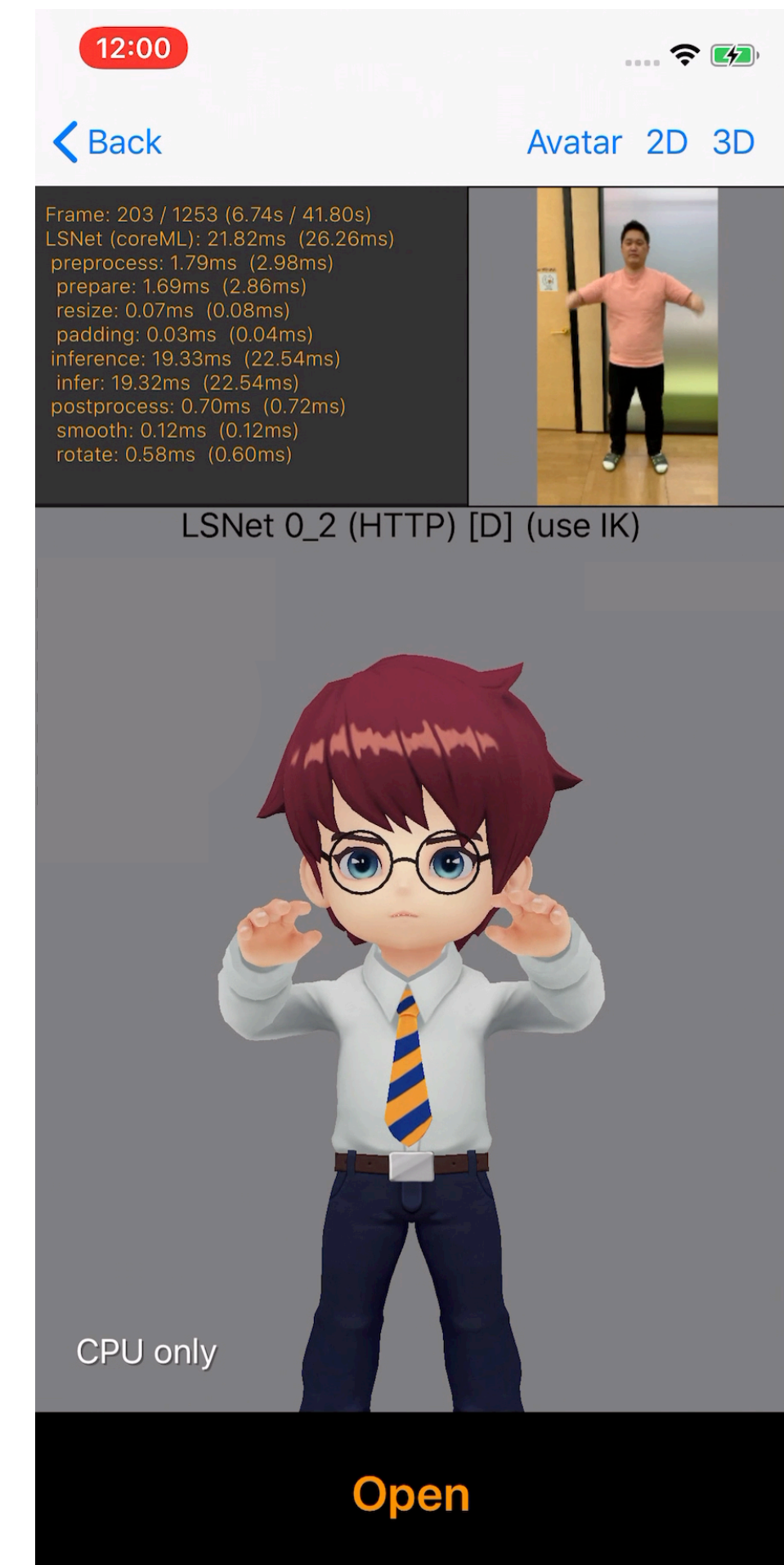
After 1year 6months



PC → Mobile + SDK

Model size: 93% ↓

Speed: x3 ↑



October, 2020

Mobile (iPhone XS) + our 3D character
Model size: 4.5Mb, 35FPS (Render + Pre/Inference/Post)

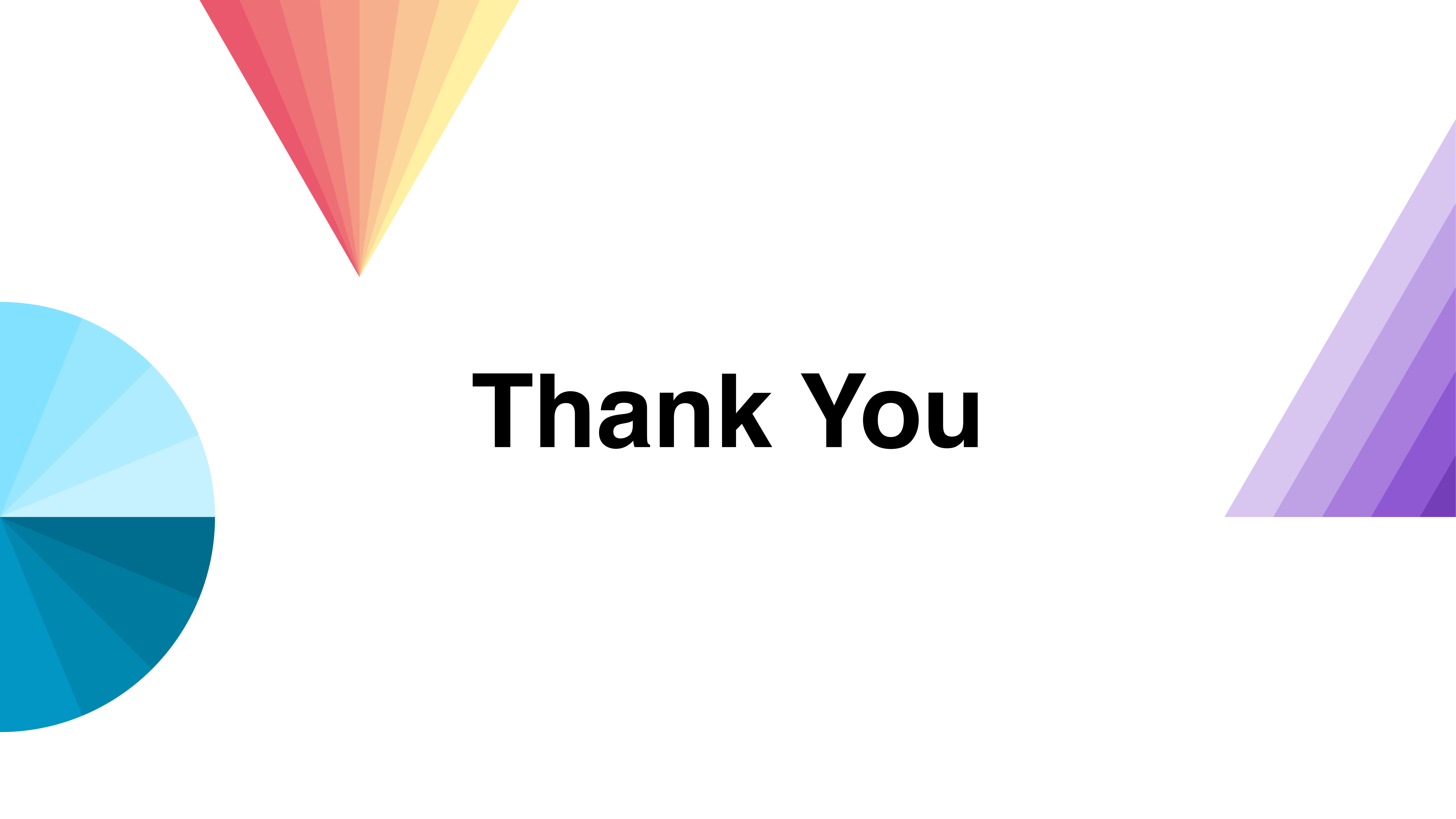
나를 따라하는 아바타: 이제 CLOVA Avatar 에서 만나요



Recruit (Intern/Regular): suntae.kim@navercorp.com



Q & A



Thank You